Informatica®

Informatica® Cloud Data Integration

Salesforce Connector

Informatica Cloud Data Integration Salesforce Connector
April 2024

© Copyright Informatica LLC 2016, 2024

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.

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.

Informatica, the Informatica logo, Informatica Cloud, and PowerCenter 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. Required third party notices are included with the product.

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

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

NOTICES

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

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

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

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

Publication Date: 2024-05-12

Table of Contents

Preface 6

Informatica Resources 6

- Informatica Documentation 6
- Informatica Intelligent Cloud Services web site 6
- Informatica Intelligent Cloud Services Communities 6
- Informatica Intelligent Cloud Services Marketplace 6
- Data Integration connector documentation 7
- Informatica Knowledge Base 7
- Informatica Intelligent Cloud Services Trust Center 7
- Informatica Global Customer Support 7

Chapter 1: Introduction to Salesforce Connector 8

Salesforce Connector assets 8

Salesforce Connector example 9

Chapter 2: Connections for Salesforce 10

Prepare for authentication 10

- Standard 10
- OAuth 10

Connect to Salesforce 11

- Before you begin 11
- Connection details 11
- Salesforce connection types 12

Firewall configuration 13

Proxy server settings 13

Connection timeout 14

Troubleshooting a Salesforce connection 14

Chapter 3: Synchronization tasks with Salesforce 15

Salesforce sources in synchronization tasks 15

- Configuring multiple Salesforce objects as the source 16
- Include archived and deleted Salesforce data in a task 17
- Rules and guidelines for Salesforce sources 17

Salesforce targets in synchronization tasks 17

- Rules and guidelines for Salesforce targets 18

Upsert task operation 18

Field mappings in Salesforce synchronization tasks 19

Advanced options in synchronization tasks 20

- Salesforce standard API 21
- Salesforce bulk API 22

Null updates to related objects. . . . 24

Chapter 4: Mappings and mapping tasks with Salesforce 25

Salesforce objects in mappings. . . . 25
Salesforce sources in mappings. . . . 25
Salesforce targets in mappings. . . . 28
Salesforce lookup objects in mappings. . . . 30
Salesforce objects in mapping tasks. . . . 30
Salesforce Sources in Mapping Tasks. . . . 31
Salesforce targets in mapping tasks. . . . 32
Salesforce lookup objects in mapping tasks. . . . 34
Custom query source type. . . . 34
Salesforce Bulk API in mappings and mapping tasks. . . . 37
Monitor the bulk job. . . . 37
Success and error files for the bulk API. . . . 37
Enable serial mode. . . . 38
Hard deletes. . . . 38
Enable primary key chunking. . . . 38

Chapter 5: Replication tasks with Salesforce. . . . 40

Replication task source properties. . . . 40
Replication task target properties. . . . 41
Load types. . . . 42
Full load. . . . 42
Incremental loads. . . . 43
Database target reset. . . . 44
Handling source and target mismatch. . . . 45
Setting the AutoAlterColumnType property. . . . 45
High precision calculations. . . . 46
Salesforce Base64 encoded body size. . . . 46
Salesforce API. . . . 46
Rules and guidelines for configuring replication tasks. . . . 46

Chapter 6: Masking tasks with Salesforce. . . . 48

Relationship reconciliation strategies. . . . 48
External ID field. . . . 48
Custom field lookup. . . . 49
Unique field lookup. . . . 49
Junction objects. . . . 51
Target owner name. . . . 51
Salesforce bulk API limits. . . . 51
Salesforce limitations in masking. . . . 52
Special handling of standard objects. . . . 53

Advanced Salesforce options. 54

Chapter 7: Common configuration. 55

Salesforce targets and IDs for related objects. 55

Object search and selection. 56

Display business names. 56

Data filters. 56

Rules and guidelines for data filters. 57

Starting tasks with Salesforce outbound messages. 57

Chapter 8: Troubleshooting. 59

Troubleshooting a Salesforce connection. 59

Troubleshooting a Salesforce synchronization task. 59

Troubleshooting a mapping task. 60

Troubleshooting a replication task. 61

Troubleshooting masking task errors. 61

Troubleshooting a bulk job. 63

Appendix A: Data type reference. 65

Salesforce datatypes and transformation datatypes. 65

Rules and guidelines for datatypes. 66

Index. 68

Preface

Use *Salesforce Connector* to learn how to read from or write to Salesforce. Learn to create a Salesforce connection, develop and run synchronization tasks, mappings, mapping tasks, replication tasks, masking tasks, dynamic mapping tasks, and data transfer tasks in Cloud Data Integration.

Informatica Resources

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

Informatica Documentation

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

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

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

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

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

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

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to Salesforce Connector

You can use Salesforce Connector to securely read data from or write data to Salesforce.

Salesforce is a cloud-based Customer Relationship Management (CRM) solution for sales teams to manage contacts and sales activities. You can use Salesforce to store and manage contacts and data of the sales activities for your organization.

You can use Salesforce connections in any Data Integration task. You can create a connection to any type of Salesforce account. You can create connections to the following Salesforce editions:

- Professional Edition
- Enterprise Edition
- Unlimited Edition

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. Salesforce Connector supports big objects for source and target transformations. Big objects are not supported for lookup and filter transformations.

Use a Salesforce connection in mappings, synchronization tasks, and mapping tasks. You can switch the mapping to advanced mode to include transformations and functions that enable advanced functionality.

You can create an OAuth type connection to allow access to Salesforce.com through its API. OAuth is a standard protocol that allows for secure API authorization. One of the benefits of OAuth is that users do not need to disclose their Salesforce credentials and the Salesforce administrator can revoke the consumer's access at any time.

Salesforce Connector assets

Create assets in Data Integration to integrate data using Salesforce Connector.

When you use Salesforce Connector, you can include the following Data Integration assets:

- Data transfer task
- Dynamic mapping task
- Mapping
- Mapping task

- Masking task
- PowerCenter task
- Replication task
- Synchronization task

*You cannot configure a Lookup transformation in a data transfer task and replication task.

Note: You cannot use a Salesforce OAuth connection for PowerCenter tasks. Use a standard Salesforce connection for PowerCenter tasks.

You can run only synchronization and mapping tasks if you configure Hosted Agent as the runtime environment in the Salesforce connection.

For more information about configuring assets and transformations, see *Mappings*, *Transformations*, and *Tasks* in the Data Integration documentation.

Salesforce Connector example

Your organization might need to migrate real-time sales opportunity information from a Salesforce system to an external system. You can use Salesforce Connector to extract data from a Salesforce system and write to the target system. The executive management team can use the external system to reconcile and analyze the data, generate report, or make decisions.

Salesforce Connector helps in rapidly synchronizing business critical data including accounts, contacts, prices, products from the applications in your organization with other key applications or databases. Sales, marketing, or any other team can move data form Salesforce to any other database or external systems so as to generate reports for easier decision making.

CHAPTER 2

Connections for Salesforce

Create a Salesforce connection to securely read data from or write data to Salesforce.

You can use a Salesforce connection in synchronization, replication, masking, PowerCenter, and mapping tasks.

Prepare for authentication

You can configure standard and OAuth authentication types to access Salesforce. Consider using OAuth authentication to connect more securely to Salesforce.
Before you configure the connection properties, you need to keep the authentication details handy based on the authentication type that you want to use.

Standard

To use a standard connection, you need the Salesforce account user name, password, and service URL. You also need your Salesforce security token to connect to Salesforce. If you do not want to use the security token, you need to add the Data Integration IP addresses to the trusted IP ranges in your Salesforce account. For more information about the list of IP address ranges used by the Secure Agent and your service, see POD Availability and Networking. For information about setting the IP address in your Salesforce account, see the Salesforce documentation.

OAuth

Create an OAuth connection that uses the OAuth 2.0 protocol to access Salesforce through the Salesforce API. OAuth is a standard protocol that allows for secure API authorization.

To create an OAuth connection, you need an OAuth refresh token. Informatica provides the SFDC OAuth 2.0 tool to generate the OAuth refresh token.

You also need the consumer key and consumer secret from your Snowflake account to generate the OAuth refresh token.

1. Download the SFDC OAuth tool from Informatica Marketplace.
2. Extract the `OAuth.zip` file.
3. Go to the `oauth\conf` folder, open the server.xml file, and update the `mystore.jks` file path to the one on your system. Save and close the file.
4. Go to `\oauth\bin` and run the command `catalina.bat start`.
5. Open `http://localhost:8090/salesforce` from the browser.

6. Enter your Salesforce user name and password to log in.
7. Enter the client ID and client secret, and click **Submit**.
   The client ID is the consumer key in Salesforce and the client secret is the consumer secret in Salesforce.

   The OAuth refresh token is generated.

Connect to Salesforce

Let's configure the Salesforce connection properties to connect to Salesforce.

By default, Salesforce connections for new organizations use the Salesforce API version 60.0. You can edit existing Salesforce connections or create new connections to use any Salesforce API version up to 60.0, except the version 58.0.

Before you begin

Before you get started, you'll need to get information from your Salesforce account based on the connection type that you want to configure.

To configure a standard connection, get the Salesforce user name, password, service URL, and security token from your Salesforce account.

To configure an OAuth connection, get the consumer key, consumer secret, and service URL from your Salesforce account. You also need to get the OAuth refresh token that is generated from the SFDC OAuth tool provided by Informatica.

Check out "Prepare for authentication" on page 10 to learn more about the authentication prerequisites.

The following video shows you how to get the information that you need from your Salesforce account and how to generate an OAuth refresh token using the SFDC OAuth tool:



Connection details

The following table describes the basic connection properties:

Property	Description
Connection Name	Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ . + -, Maximum length is 255 characters.
Description	Description of the connection. Maximum length is 4000 characters.

Property	Description
Type	Salesforce
Runtime Environment	The name of the runtime environment where you want to run tasks. Select a Secure Agent, Hosted Agent, or serverless runtime environment. Hosted Agent doesn't apply to mappings in advanced mode.

Salesforce connection types

You can configure standard and OAuth connection types to access Salesforce.
Select the required connection type and then configure the connection-specific parameters.

Standard

A Salesforce standard connection requires at a minimum your Salesforce account user name, password, and service URL.
The following table describes the basic connection properties for standard connection:

Property	Description
User Name	The user name to connect to the Salesforce account.
Password	The password to connect to the Salesforce account.
Service URL	URL of the Salesforce service. For example: `https://login.salesforce.com/services/Soap/u/60.0` By default, Salesforce connections for new organizations use version 60.0 of the Salesforce API. You can use any Salesforce API version up to 60.0, except the version 58.0. Maximum length is 100 characters. If you edit the service URL for an existing standard connection, you need to re-enter the password and security token.

Advanced settings

The following table describes the advanced connection properties for standard connection:

Property	Description
Security Token	Security token generated from the Salesforce application. **Note:** If you don't enter the Salesforce security token in this field, you need to add the Data Integration IP addresses to the trusted IP ranges in your Salesforce account. Otherwise, the connection might fail. For more information about the list of IP address ranges, see "Standard" on page 10.
Bypass proxy server settings defined for the Secure Agent	Bypasses the proxy server settings defined for the Secure Agent in the Secure Agent Manager and directly connects to Salesforce. If you do not select this option, the Salesforce connection uses the proxy server setting defined for the Secure Agent to connect to Salesforce.

OAuth connection

OAuth connection requires your Salesforce consumer key, secret, and refresh token. When you use OAuth connection, you need to use the OAuth 2.0 protocol.

The following table describes the basic connection properties for OAuth connection:

Property	Description
OAuth Consumer Key	The consumer key to generate a refresh token.
OAuth Consumer Secret	The consumer secret to generate a refresh token.
OAuth Refresh Token	The refresh token that you generated using the SFDC OAuth 2.0 tool. For more information about how to generate the OAuth refresh token, see the *Before you begin* section.
Service URL	URL of the Salesforce service endpoint. For example: `https://login.salesforce.com/services/Soap/u/60.0` You can use any Salesforce API version up to 60.0, except the version 58.0. Maximum length is 100 characters. When you edit the service URL for an existing OAuth connection, you need to re-enter the consumer key, consumer secret, and refresh token.

Advanced settings

The **Service End Point** and **OAuth Access Token** fields do not apply to OAuth authentication.

Firewall configuration

If your organization passes data through a firewall, you need to configure the firewall to allow access to Salesforce.

If you cannot connect to Salesforce servers and receive a connection error, contact your network administrator to allow access to Salesforce servers.

For more information, see the following Knowledge Base article: Firewall rules

Note: IP addresses for Salesforce servers might change. For the latest information about the server IP addresses of Salesforce, see the Salesforce documentation.

Proxy server settings

If your organization uses an outgoing proxy server to connect to the Internet, the Secure Agent connects to Informatica Intelligent Cloud Services through the proxy server.

You can configure the Secure Agent to use the proxy server on Windows and Linux. You can use the unauthenticated or authenticated proxy server. The proxy settings applies to connections used in mappings and in mappings in advanced mode.

To configure the proxy settings for the Secure Agent, use one of the following methods:

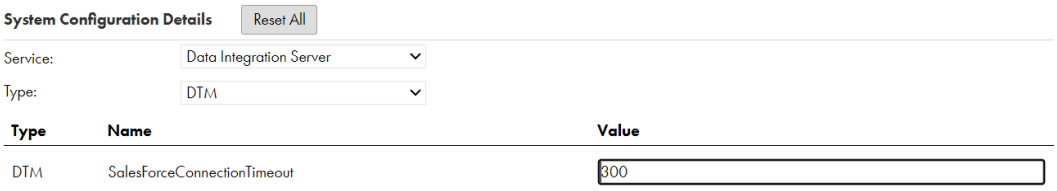
- Configure the Secure Agent through the Secure Agent Manager on Windows or shell command on Linux.

  For instructions, see "Configure the proxy settings on Windows" or "Configure the proxy settings on Linux" in *Getting Started* in the Data Integration help .
- Configure the JVM options for the DTM in the Secure Agent properties. For instructions, see the Proxy server settings Knowledge Base article.

To configure proxy settings for the serverless runtime environment, see "Using a proxy server" in *Runtime Environments* in the Administrator help.

Connection timeout

You can set the `SalesForceConnectionTimeout` property for the DTM in the Secure Agent configuration properties to the duration, in seconds, that the connection waits for a response to its requests from the Salesforce web service. If the web service doesn't respond within the specified timeout period, the request times out.
The following image shows the configured `SalesForceConnectionTimeout` property for the Secure Agent:



Troubleshooting a Salesforce connection

If you encounter errors when you configure a Salesforce connection, you might have to enter the Salesforce security token in the Salesforce connection properties.

If the security token is required and the **Security Token** field in the Salesforce connection is empty or not valid, the following error message appears when you test or create a connection:

```
The login to Salesforce.com failed with the following message -
LOGIN_MUST_USE_SECURITY_TOKEN:
```

Go to the Salesforce web site to get the security token. To avoid adding the security token to the connection details, you can also add Data Integration IP addresses to the trusted IP ranges in your Salesforce account.

CHAPTER 3

Synchronization tasks with Salesforce

Use the Synchronization task to synchronize data between a source and target.

You can configure a synchronization task using the Synchronization Task wizard.

When you create a task, you can associate it with a schedule to run it at specified times or on regular intervals. Or, you can run it manually. You can monitor tasks that are currently running in the activity monitor and view logs about completed tasks in the activity log.

Salesforce sources in synchronization tasks

You can use a single object, multiple related objects, or saved query in a synchronization task.

You configure Salesforce source properties on the **Source** page of the Synchronization Task wizard.

The following table describes the Salesforce source properties:

Property	Description
Connection	Name of the source connection.
Source Type	Select Single, Multiple, or Saved Query.
Source Object	Name of the source object.
Display technical names instead of labels	Displays technical names instead of business names.
Display source fields in alphabetical order	Displays source fields in alphabetical order instead of the order returned by the source system.
Include archived and deleted rows in the source	Includes archived and deleted rows in the source. By default, the agent returns active rows.

You can add multiple objects that have an explicit relationship defined in Salesforce. For example, if you use the Opportunity object as a source, you can add the related Account as well. You can also add the Master Record object because it is related to the Account object.

Configuring multiple Salesforce objects as the source

You can configure multiple Salesforce objects as the source of a synchronization task.

1. On the **Source** page, select the Salesforce connection.

   To create a connection, click **New**. To edit a connection, click **View**, and in the View Connection dialog box, click **Edit**.

2. Select **Multiple**.

   The Source Objects table displays. The Action column lists the actions that you can perform on each row. The Source Object column shows each selected Salesforce source. The Relationship Path column shows the relationship between the source and the primary Salesforce source.

3. Click **Add** to add the primary Salesforce source.

4. In the **Select Source Object** dialog box, select the primary Salesforce source you want to use, and then click **OK**.

   The dialog box displays up to 200 objects. If the objects that you want to use do not display, enter a search string to reduce the number of objects that display.

   Note: For Salesforce, only Salesforce objects that can be queried appear in the Objects to Replicate area. If an object does not appear, contact the Salesforce administrator.

5. To add a source related to the primary source, highlight the primary source in the Source Objects table, and then click **Add**.

   The **Add Related Objects** dialog box displays. The dialog box shows the available relationships that have not been associated with the selected source object. The synchronization Task Wizard selects the object based on the selected relationship.

6. Select a relationship type and click **OK**.

   The selected relationship path displays in the Source Objects table. To remove a source, click the **Remove** icon in Action column for the source.

7. To add additional related sources, select the source you want to use and click **OK**. Then select another relationship type and click **OK**.

8. To display technical names instead of business names for some connection types, select **Display technical field names instead of labels**.

9. To display source fields in alphabetical order, click **Display source fields in alphabetical order**.

   By default, fields appear in the order returned by the source system.

10. If you want the synchronization task to read historical data in Salesforce sources, including archived and deleted rows, select **Include Archived and Deleted Rows in the Source**.

    By default, the synchronization task reads only current source rows in selected Salesforce sources and ignores archived and deleted rows.

11. To preview source data, select the source in the Source Objects table. If preview data does not appear automatically, click **Show Data Preview**.

    The Data Preview area shows the first ten rows of the first five columns in the source. It also displays the total number of columns in the source.

    The Data Preview area does not display certain Unicode characters as expected. If the data contains binary data, the Data Preview area shows the following text:

    ```
    BINARY DATA
    ```

12. To preview all source columns in a file, select the source in the Source Objects table and click **Preview All Columns**.

    The file shows the first ten rows of the source.

13. Click **Next**.

Include archived and deleted Salesforce data in a task

When you use Salesforce sources in a task with the Salesforce standard API and Bulk API, you can fetch deleted and archived source data by selecting the Include archived and deleted rows in the source field. By default, the synchronization task reads only current source rows in selected Salesforce sources and ignores archived and deleted rows.

When you select this option, the synchronization task fetches deleted and archived Salesforce data from the source.

The delete task operation for the Salesforce Bulk API is available for Salesforce API version 39.0 and later.

Rules and guidelines for Salesforce sources

Consider the following rules and guidelines when you configure a Salesforce source:

- A task with multiple Salesforce source objects and a flat file target fails if the agent cannot replicate all source objects when you run the task.
- All objects must be available through the same source connection. All Salesforce objects in a multiple-object source must have a predefined relationship in Salesforce.
- The synchronization Task wizard removes a user-defined join when you change the source connection from database to flat file or Salesforce.
- When you create a connection to a Salesforce source or target, the Data Integration caches the connection metadata. If the connection metadata changes while you are creating the connection, you must log out and log back in to refresh the connection metadata.
- A synchronization task fails if the lookup field in a Salesforce object has the Text Area (Long) datatype.
- A saved query fails to parse if you use an ORDER BY clause in a nested query.
- You cannot preview data for a saved query with unicode filter.
- If you enable the Salesforce Bulk API in a synchronization task, you cannot use a custom query as source.

Salesforce targets in synchronization tasks

You can use a single object as a target in a synchronization task.

The target connections that you can use depend on the task operation that you select for the task.

The following table describes the Salesforce target properties:

Property	Description
Connection	Name of the target connection.
Target Object	Name of the target object.

Property	Description
Display technical names instead of labels	Displays technical names instead of business names.
Display target fields in alphabetical order	Displays target fields in alphabetical order instead of the order returned by the target system.

Rules and guidelines for Salesforce targets

Consider the following rules and guidelines for Salesforce targets:

- The source must provide non-null values for required fields in a Salesforce target object.
- When you use related objects in field mappings, you need to specify the External ID for each related object.
- A task might lose the least significant portion of numerical data in a Salesforce target field when the data uses most of the maximum precision and scale specified for the data type of the field.
  For example, when you try to insert 65656565656565.6969 into a Salesforce field with data type Decimal (14, 4), the task inserts 65,656,565,656,565.6950 instead. When you try to insert 123456789123456789 into a Salesforce field with data type Decimal (18, 0), the task inserts 123,456,789,123,456,784.
- A task might round data unexpectedly into a Salesforce target field when the data uses most of the maximum precision and scale specified for the data type of the field.
  For example, when you try to insert 555555555565.8855 into a Salesforce field with data type Decimal (18, 0), the task inserts 555555555566 instead. However, when you manually enter the data in Salesforce, Salesforce rounds the data as 555555555565. When you try to insert 12345678923456.3399 into a Salesforce field with data type Decimal (14, 4), the task inserts 12,345,678,923,456.3400 instead.
- When you use an Oracle source with a Salesforce target, if the source contains fields with the Number data type, change the field type to numeric. Values for fields with the Number data type do not load correctly in Salesforce. You can change the type on the **Field Mapping** page of the synchronization Task wizard. Alternatively, if you have many fields with the Number data type, you can add the `oracle.use.varchar.for.number` custom property for the Secure Agent.

Upsert task operation

When you configure a synchronization task to perform upserts on a Salesforce target, configure the upsert field on the Schedule page of the Synchronization Task wizard. You can use an ID field for standard objects. You can use external ID field or any field with the idLookup field property enabled for standard and custom objects. Ensure that you include the upsert field in the field mappings for the task.

Note: The task fails if the Salesforce user account defined does not have creatable or updatable permissions on the external ID field or any field with the idLookup field property enabled. When you configure the upsert task operation with an external ID or idLookup field that is write-protected, ensure that the data exists.

Field mappings in Salesforce synchronization tasks

You can map source columns to target columns on the Field Mapping page of the Synchronization Task wizard. You must map at least one source column to a target column. You can map columns with compatible data types or create field expressions to convert data types appropriately.

Based on the operation, the synchronization task requires certain fields in the field mapping. By default, the synchronization task maps the fields with similar names. When you configure the field mapping, ensure that the required fields remain mapped. If you do not map the required fields, the synchronization task fails.

The following table shows the required fields for each task operation and target type:

Required Field	Task Operations	Description
ID	Delete Update	Map the ID column to enable the synchronization task to identify records to delete or update in a Salesforce target.
Upsert Field	Upsert	Configure and map the upsert field to enable the synchronization task to identify records to upsert in a Salesforce target. The upsert field can be an external ID field or a field with the idLookup field property enabled.
Non-null fields	Insert Update Upsert	Map all fields that cannot be null in Salesforce.

When you configure field mappings, you can also perform the following tasks:

- Map a column with similar name.
- Edit field data types.
- Add a mapplet to the field mapping.
- Create field expressions to transform data.
- Create lookups.
- Validate expressions defined in the field mapping section.

If you included multiple source objects in the task, you can select each source object in the Sources field to display the fields for the selected object. Or, you can view all source object fields. When displaying all source object fields, the Sources table displays field names grouped by source object. You can place the cursor over the Status icon for a source field to determine the Label and API names of Salesforce source fields. Label names appear in the Salesforce UI. API names are the names required by the Salesforce API.

To configure external IDs for related Salesforce objects, use **Related Objects** to select the external ID for each applicable related object. You do not need to specify the external ID for a related object if you do not want to update changes for that related object. If you do not specify the external ID, the synchronization task requires that the source provide the Salesforce ID to uniquely identify records in each related object.

Advanced options in synchronization tasks

When you configure a synchronization task in Salesforce, you can use the following advanced source and target options:

Advanced Salesforce Option	Description
Allow Null Updates to Target	Salesforce targets only. Indicates if null values are allowed to replace existing values in the target. Select True to allow null values to replace existing values in the target. Default is False.
Salesforce API	API used to process Salesforce source or target data. Select one of the following options: - Standard API. Uses the Salesforce standard API to process Salesforce data. - Bulk API. Uses the Salesforce Bulk API to process Salesforce data.
Target Batch Size	For loading to Salesforce targets using the standard API. The maximum number of records to include in each query that writes to the Salesforce target. Enter a number between 1 and 200. To process multiple upserts on a particular row in the same query, set the batch size to 1. Default is 200.
Create the Success File	For loading to Salesforce targets using the standard API. Creates a success file for a standard API task.
Assignment Rule Selection	For loading to Salesforce Case, Lead, or Account target objects using the standard API. Assignment rule to reassign attributes in records when you insert, update, or upsert records: - **None**. Select to use no assignment rule. Default is None. - **Default**. Select to use the default assignment rule set for the organization. - **Custom**. Applicable only to Case and Lead objects. Select to specify and use a custom assignment rule.
Monitor the Bulk Job	For loading to Salesforce targets using the Bulk API. Monitors a Bulk API job to provide accurate session statistics for each batch. Generates a Bulk API error file with row-level details based on information provided by the Salesforce Bulk API. Without monitoring, the All Jobs page and session log cannot provide information about batch processing or create success and error files. Monitoring requires additional Bulk API calls.
Create the Success File	For loading to Salesforce targets using the Bulk API with monitoring enabled. Creates a success file with row-level details based on information provided by the Salesforce Bulk API.
Enable Serial Mode	For loading to Salesforce targets using the Bulk API. Salesforce loads Bulk API batches to the target serially. By default, Salesforce loads Bulk API batches in parallel.
Enable Hard Delete	For loading to Salesforce targets using the Bulk API with the Delete task operation. Permanently deletes target rows. Deleted rows cannot be recovered.

Advanced Salesforce Option	Description
Enable PK Chunking	Use to extract from Salesforce sources when you use the Bulk API. Enables primary key chunking to optimize performance while extracting from large data sets. Salesforce splits the data set into a number chunks based on the record ID, creates multiple queries to extract data, and combines the result. Salesforce supports primary key chunking for custom objects and certain standard objects. For more information about supported objects for primary key chunking, see the Salesforce documentation.
PK Chunking Size	The number of records in a chunk. Default is 100,000. The maximum value is 250,000. Applicable only if you select Enable PK Chunking.
PK Chunking startRow ID	The record ID from which you want to chunk the data set. By default Salesforce applies chunking from the first record. Applicable only if you select Enable PK Chunking.
PK Chunking Parent Object	Not applicable.

Salesforce standard API

You can use the Salesforce standard API to read data from Salesforce sources and write data to Salesforce targets. Use the standard API to process a normal amount of Salesforce data and standard reporting on the results of the standard API load.

Target batch size for the standard API

When you use the Salesforce standard API to write to Salesforce targets, you can configure the target batch size used to write data to Salesforce.

The target batch size determines the maximum number of records to include in each query that writes to the Salesforce target. Salesforce allows up to 200 records for each query. If you enter a value higher than 200, each query includes only 200 rows. Default is 200.

You might use a smaller batch size for upserts because you cannot update the same row more than once in a single query. To process multiple upserts on a particular row in the same query, set the batch size to 1.

Salesforce limits the number of queries you can make in a 24-hour period.

Success and error files for the standard API

When you use the Salesforce standard API to write to Salesforce targets, the synchronization task creates a Salesforce error file by default. You can configure the task to create a Salesforce success file. You can generate two Salesforce success files, one each with UTC timestamp and Secure Agent local timestamp.

The Salesforce success file contains one row for each successfully processed row. Each row contains the row ID, data, and one of the following task operations: Created, Updated, Upserted, or Deleted. Use the success file to track rows that are created if you need to roll back the operation.

The Salesforce error file contains one row for each row that is not written to the target. The error file contains an entry for each data error. Each log entry contains the values for all fields of the record and the error message. Use this file to understand why records did not load into the Salesforce target.

The following table describes the location and naming convention for the standard API success and error files:

File Type	Location	Naming Convention
Standard API success file	<Secure Agent installation directory>\apps\Data_Integration_Server\data\success	s_dss_<TaskID>_TimeStamp_standard_success.csv
Standard API error file	<Secure Agent installation directory>\apps\Data_Integration_Server\data\error	s_dss_<TaskID>_TimeStamp_standard_error.csv

To generate a success file with Secure Agent local timestamp:

1. Navigate to the **Schedule** page of the Synchronization Task wizard.
2. In the **Advanced Salesforce Options** area, for the **Salesforce API**, select **Standard API**.
3. Select **Create the Success File**.
4. Save your changes.

To generate the additional success file with UTC timestamp:

1. Navigate to the **Schedule** page of the Synchronization Task wizard.
2. In the **Advanced Salesforce Options** area, for the **Salesforce API**, select **Standard API**.
3. Select **Create the Success File**.
4. Save your changes.
5. Click Administrator > Runtime Environments and select an agent.
6. Click **Edit.**
7. Select Type as DTM under **System Configuration Details**.
8. Set the JVMOption1 as `-DSFDCCreateSuccessErrorFileFromParams=true`.
9. Click **Save** to save the changes.

Salesforce bulk API

You can use the Salesforce Bulk API to read data from Salesforce sources and write data to Salesforce targets. Use the Bulk API to process large amounts of Salesforce data while generating a minimal number of API calls.

When you use the Salesforce Bulk API, Salesforce restricts the size of each batch to 10 MB of data or 10,000 records in CSV format. For example, if you want to read 10 million records, the records are split into 1000 batches and each batch contains 10000 records. If the size of the 10000 records exceeds 10 MB, the batches are split based on the size instead of the number of records.

When the synchronization task creates a batch, it adds any required characters to format the data, such as adding quotation marks around text.

You can monitor jobs that use the Bulk API to write to Salesforce targets. When you monitor a Bulk API target job, the synchronization task can create success and error files for row-level information. The synchronization task can also load batches at the same time or serially.

Monitor the bulk job

When you use the Bulk API to write to Salesforce targets, you enable the task for monitoring. With monitoring enabled, the synchronization task requests the status of each batch from the Salesforce service. The synchronization task repeats the request every 10 seconds until all batches complete. The synchronization task writes the responses from the Salesforce service to the All Jobs page and the session log. With monitoring enabled, the synchronization task also generates a Bulk API error file.

By default, the synchronization task permits monitoring for Bulk API jobs. You can configure the task to run without monitoring. Without monitoring, the All Jobs page and the session log contain information about batch creation, but do not contain details about batch processing or accurate job statistics.

Note: The synchronization task performs additional API calls when monitoring a Bulk API job. To reduce the number of API calls the synchronization task makes, do not monitor the job. For more information about batch processing, use the Job and Batch IDs from the session log to access Salesforce statistics.

Success and error files for the bulk API

When you monitor a Bulk API target job, the synchronization task generates a Bulk API error file by default. You can configure the task to create a Bulk API success file. Success and error files are CSV files that contain row-level details provided by the Salesforce service.

The Bulk API success and error files include the job ID, batch ID, Id, success, created, and error message information.

The following table describes the location and naming convention for the Bulk API success and error files:

File Type	Location	Naming Convention
Bulk API success file	`<Secure Agent installation directory>\apps\Data_Integration_Server\data\success`	`s_dss_<TaskID>_TimeStamp_bulk_success.csv`
Bulk API error file	`<Secure Agent installation directory>\apps\Data_Integration_Server\data\error`	`s_dss_<TaskID>_TimeStamp_bulk_error.csv`

To generate a success file:

1. Navigate to the **Schedule** page of the Synchronization Task wizard.
2. In the **Advanced Salesforce Options** area, for the `Salesforce API`, select **Bulk API**.
3. Select **Monitor the Bulk Job**.
4. Select **Create the Success File**.
5. Save your changes.

Enable serial mode

When you use the Bulk API to load data to Salesforce, you can configure the task to perform a parallel load or a serial load. By default, it performs a parallel load.

In a parallel load, Salesforce writes batches to targets at the same time. Salesforce processes each batch whenever possible. In a serial load, Salesforce writes batches to targets in the order it receives them. Salesforce processes the entire content of each batch before processing the next batch.

Use a parallel load to increase performance when you do not need a specific target load order. Use a serial load when you want to preserve the target load order, such as during an upsert load.

Hard deletes

When you use the Bulk API, you can configure the task to permanently delete data from Salesforce targets.

When you use a Bulk API task to delete data from Salesforce targets, the synchronization task copies the deleted rows to the recycle bin. You can retrieve deleted data within a certain amount of time, but it also requires additional space from the hard disk.

With a hard delete, the synchronization task bypasses the recycle bin. You cannot recover data if you delete the data with the hard delete option.

Enable primary key chunking

Enable primary key chunking to increase performance when you extract data from large tables.

When you use the Bulk API to extract data from Salesforce, you can enable primary key chunking. By default, the Bulk API does not use primary key chunking.

When you enable primary key chunking, the Bulk API splits the data set into multiple chunks based on the record ID and creates extract queries for each chunk. The Bulk API combines the data when all the extract queries are complete.

Salesforce supports primary key chunking for custom objects and certain standard objects. For more information about objects that support primary key chunking, see the Salesforce documentation.

Note: You can enable primary key chunking when your Salesforce connection uses version 32 or higher of the Salesforce API. The default chunk size is 100000.

Null updates to related objects

You can perform null updates to related Salesforce objects. The results of a null update depend on the Salesforce API that you use.

When you use the standard API to update a null value to a field in a related object, Salesforce dissociates the related object.

When you use the Bulk API to update a null value to a field in a related object, Salesforce updates the field with the null value.

CHAPTER 4

Mappings and mapping tasks with Salesforce

When you configure a mapping, you describe the flow of data from the source to the target.

A mapping defines reusable data flow logic that you can use in mapping tasks. When you create a mapping, you define the Source, Target, and Lookup transformations to represent a Salesforce object. Use the Mapping Designer in Data Integration to add the Source, Target, or Lookup transformations in the mapping canvas and configure the Salesforce source, target, and lookup properties.

In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality.

You can define the following objects in the mapping:

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

Validate and run the mapping to extract the Salesforce data and load it to a relational or flat file target.

You can create an input parameter for logical aspects of a data flow. For example, you might use a parameter in a filter condition and a parameter for the target object. Then, you can create multiple tasks based on the mapping and write different sets of data to different targets. You could also use an input parameter for the target connection to write target data to different Salesforce accounts.

Salesforce objects in mappings

When you create a mapping, you can configure a Source or Target transformation to represent a Salesforce object.

Salesforce sources in mappings

In a mapping, you can configure a Source transformation to represent a single Salesforce source or multiple Salesforce sources.

Specify the name and description of the Salesforce source. Configure the source and advanced properties for the source object.

The following table describes the Salesforce source properties that you can configure in a Source transformation:

Property	Description
Connection	Name of the source connection.
Source Type	Type of the source object. Select Single Object, Multiple Objects, or Parameter.
Object	Name of the source object for the mapping. You can specify a custom query for a source object.
Filter	Adds conditions to filter records. Configure a simple or an advanced filter. You cannot use the LIMIT clause in an advanced filter. To limit the number of rows, specify the row limit in the advanced source properties.
Sort	Not applicable for a Salesforce connection.
Include archived and deleted rows in the source	Includes archived and deleted source rows. By default, the agent returns active rows.

The following table describes the Salesforce advanced source properties that you can configure in a Source transformation:

Advanced Property	Description
Row Limit	The maximum number of rows the agent processes. Select All Rows to process all records, or Specify number of rows to process specific number of rows.
Salesforce API	Salesforce API to read source data. Select one of the following options: - Standard API - Bulk API - Bulk API 2.0 You must use an OAuth connection if you select the Bulk API 2.0 option to read data.
Enable PK Chunking	Select to enable the primary key chunking. When you enable primary key chunking, the Bulk API splits the data set into multiple chunks based on the record ID and creates extract queries for each chunk. Doesn't apply to Salesforce Bulk API 2.0.
PK Chunking Size	The size of data in a chunk or batch size. For example, if you want to upload a 10 MB file, you can upload the file in fragments, that is, 1MB at a time. When you mention 1MB as the chunk size,1 MB of data is transferred at a time until the complete transfer takes place.
PK Chunking startRow ID	The row ID from where the chunking starts.
SOQL Filter Condition	SOQL condition to filter source data. You cannot use the LIMIT clause in the filter condition. To limit the number of rows, specify the row limit in the advanced source properties. If you configure the filter under Salesforce source properties as well as the SOQL Filter Condition, then the SOQL Filter Condition overrides the filter under Salesforce source properties.

Advanced Property	Description
Maximum Rows Per Request	The maximum number of records to retrieve per set of results for the query. The request is still subject to the size limits. Default is 0. Applies to Salesforce Bulk API 2.0 only.
Tracing Level	Amount of detail that appears in the log for this transformation. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.

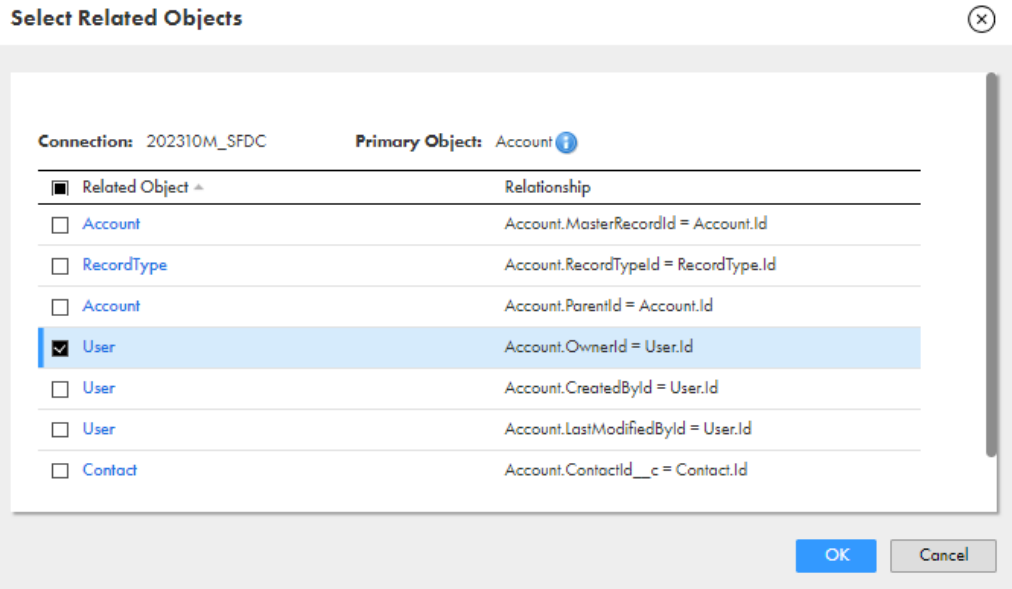
Adding multiple source objects

When you configure a Source transformation to read from multiple Salesforce objects, you can select related objects based on existing relationships in Salesforce.

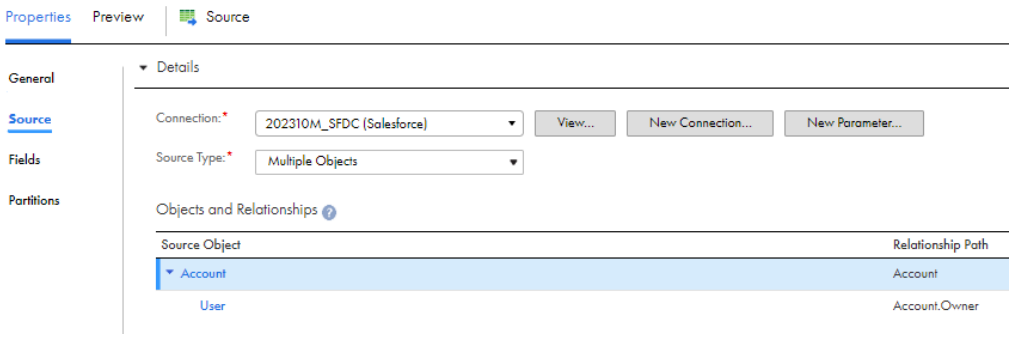
1. On the Source page in the mapping, select the Salesforce connection.
2. Select **Multiple Objects** as the **Source Type**.
3. In the **Objects and Relationships** section, click **Add Source Objects** from the menu.
4. In the **Select Source Object** window, select the source object that you want to add, and then click **OK**.
5. To add related objects, select the source object, right-click menu, and then click **Add Related Objects**.

6. In the **Select Related Object** window, select the related object that you want to add, and then click **OK**.

   For example, the following image shows the list of related objects that have existing relationships with the Account parent object:



   The following image shows the selected parent-child objects in the Source transformation:



Salesforce targets in mappings

In a mapping, you can configure a Salesforce object as the Target transformation to insert data to Salesforce.

Specify the name and description of the Salesforce target. Configure the target and advanced properties for the target object.

The following table describes the Salesforce target properties that you can configure in a Target transformation:

Property	Description
Connection	Name of the target connection.
Target Type	Type of the target object. Select Single Object or Parameter.

Property	Description
Object	Name of the target object. Target object for a single target or primary target object for multiple targets.
Operation	Target operation. Select Insert, Update, Upsert, Delete, or Data Driven.

The following table describes the Salesforce target advanced properties that you can configure in a Target transformation:

Advanced Property	Description
Max Rows per Batch	The maximum number of records to include in each query that writes to the Salesforce target. You can use the following Salesforce API when you load data to the Salesforce target: - Standard API. Enter a number between 1 and 200. To process multiple upserts on a particular row in the same query, set the batch size to 1. Default is 200. - Bulk API. Enter a number between 201 and 10,000. Default is 10,000. **Note:** To use Max Rows Per Batch property for Bulk API, add **SalesforceBulkBatchSizeOverride=true** custom property in the **Advanced Session Properties** on the **Runtime Options** tab of the mapping task.
Set to Null	Indicates if null values are allowed to replace existing values in the target. Select True to allow null values to replace existing values in the target. Default is False.
Use SFDC Error File	Creates an error file with the Salesforce error logs when the task runs. By default, the Secure Agent does not generate the error log files. Select this property so that the Secure Agent generates the error logs in the following directory: `<Secure Agent installation directory>/apps/Data_Integration_Server/data/error`.
Use SFDC Success File	Creates a success file with the Salesforce success logs when the task runs. By default, the Secure Agent does not generate the success log files. Select this property so that the Secure Agent generates the success logs in the following directory: `<Secure Agent installation directory>/apps/Data_Integration_Server/data/success`.
Success File Location	Overrides the default success file directory with the path you specify. Applicable when you select the Use SFDC Success File property. To override, specify the location in the Secure Agent machine where you want the agent to generate the success log file.
Salesforce API	Salesforce API to write target data. Select any one of the options: - Standard API - Bulk API - Bulk API 2.0
Monitor Bulk	Monitors a Salesforce Bulk API target.
Monitor Bulk 2.0	Monitors a Salesforce Bulk API 2.0 target.

Advanced Property	Description
Enable Serial Mode	Loads Salesforce Bulk API batches to the target serially. By default, Salesforce loads Bulk API batches in parallel. Applies to Salesforce Bulk API only.
Hard Delete	For loading to Salesforce targets using the Bulk Delete operation. Permanently deletes target rows. Deleted rows cannot be recovered. Hard Delete option is not applicable for standard API. It applies to Bulk API and Bulk API 2.0 only.
Forward Rejected Rows	Forwards rejected rows to the next session.

Salesforce lookup objects in mappings

You can retrieve data from a Salesforce lookup object based on the specified lookup condition.

When you configure a lookup in Salesforce, you select the lookup connection and lookup object. You also define the behavior when a lookup condition returns more than one match.

Note: Salesforce lookup works only with the Salesforce Standard API.

For an uncached lookup, you cannot configure logical operators, such as <, >, <=, and >= in the lookup condition.

The following table describes the Salesforce lookup object properties that you can configure in a Lookup transformation:

Lookup Object Properties	Description
Connection	Name of the lookup connection.
Source Type	Type of the source object. Select Single Object or Parameter.
Lookup Object	Name of the lookup object for the mapping.
Multiple Matches	Behavior when the lookup condition returns multiple matches. You can return first row, last row, any row, all rows, or report an error. If you choose all rows and there are multiple matches, the Lookup transformation is an active transformation. If you choose any row, the first row, or the last row and there are multiple matches, the Lookup transformation is a passive transformation.

The Data Integration service uses case sensitive approach when performing lookups. For Salesforce lookups, the case sensitive comparison depends on the Salesforce support.

Salesforce objects in mapping tasks

When you configure a mapping task, you can configure advanced properties for Salesforce sources and targets.

Salesforce Sources in Mapping Tasks

For Salesforce source connections used in template-based mapping tasks, you can configure advanced properties in the Sources page of the Mapping Task wizard.

You can configure the following properties:

Property	Description
Connection	Name of the source connection.
Source Type	Type of the source connection. Select Single, Multiple, or Query.
Source Object	Name of the source object. You can define a custom query to use as the source object.
Query Options	You can choose from the following options: - Filter. Filter condition filters data in Salesforce. Configure a simple or an advanced filter. You cannot use the LIMIT clause in an advanced filter. To limit the number of rows, specify the row limit in the advanced source properties - Display technical names instead of labels. Displays technical names instead of business names. - Display source fields in alphabetical order. Displays source fields in alphabetical order instead of the order returned by the source system.
Related Objects	Includes related objects in the task. You can join objects with existing relationships for Salesforce connection type.

You can configure the following advanced properties:

Property	Description
SOQL Filter Condition	Enter a filter condition to filter Salesforce source records. You cannot use the LIMIT clause in the filter condition. To limit the number of rows, specify the row limit in the advanced source properties. If you configure the filter under Query Options as well as the SOQL Filter Condition, the SOQL Filter Condition overrides the filter under Query Options.
CDC Time Limit	Time period, in seconds, that the agent reads changed Salesforce data. When you set the CDC Time Limit to a non-zero value, the agent performs a full initial read of the source data and then captures changes to the Salesforce data for the time period you specify. Set the value to -1 to capture changed data for an infinite period of time. Default is 0.
Flush Interval	Interval, in seconds, at which the agent 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 agent ignores this value.
CDC Start Timestamp	Start date and time for the time period. The agent extracts data added or modified after this time. Must be in the format YYYY-MM-DDTHH:MI:SS.SSSZ. You can also use the $Paramstart mapping variable in a parameter file to specify the CDC start time.
CDC End Timestamp	End date and time for the time period. The agent extracts data added or modified before this time. Must be in the format YYYY-MM-DDTHH:MI:SS.SSSZ. You can also use the $Paramend mapping variable in a parameter file to specify the CDC end time.
Row Limit	The maximum number of rows the agent processes. Default is 0. The default value indicates that there is no row limit, and the agent processes all records.

Property	Description
Use queryAll	Runs a query that returns all rows, including active, archived, and deleted rows that are available in the recycle bin. Otherwise, the agent returns active rows. The Use queryAll property for Bulk API is available for Salesforce API version 39.0 and later. The Secure Agent ignores this property when you configure the session to perform change data capture.
Use SystemModstamp for CDC	Uses the SystemModstamp as the time stamp for changed records in Salesforce. Otherwise, the agent uses the LastModifiedDate time stamp to identify changed records in Salesforce. Default is to use the LastModifiedDate time stamp.
Enable Bulk Query	Uses the Salesforce Bulk API to read Salesforce source data. By default, the Secure Agent uses the standard Salesforce API.
Enable Bulk 2.0 Query	Uses the Salesforce Bulk API 2.0 to read large amount of data from Salesforce.

Note: You must configure only one CDC time attribute, either CDC Time Limit or CDC Start and End Timestamp. If you configure both attributes, you might encounter unexpected results.

Salesforce targets in mapping tasks

For Salesforce target connections used in template-based mapping tasks, you can configure advanced properties in the Targets page of the Mapping Task wizard.

You can configure the following properties:

Property	Description
Connection	Name of the target connection.
Target Object	Type of the target object.
Filter	Filter condition filters data in Salesforce.
Display technical names instead of labels	Displays technical names instead of business names.
Display target fields in alphabetical order	Displays target fields in alphabetical order instead of the order returned by the target system.

You can configure the following advanced properties:

Advanced Property	Description
Treat Insert as Upsert	Upserts any records flagged as insert. By default, the agent treats all records as insert.
Treat Update as Upsert	Upserts any records flagged as update. Select this property when you use the Update Strategy transformation in the mapping. Select the Treat Source Rows As session property to flag records as update.

Advanced Property	Description
Max Batch Size	Maximum number of records the agent writes to a Salesforce target in one batch. Default is 200 records. This property is not applicable in Bulk API target sessions.
Set Fields to Null	Replaces values in the target with null values from the source. By default, the agent does not replace values in a record with null values during an update or upsert operation.
Use SFDC Error File	Generates the error log files. By default, the agent does not generate the error log files. To generate an error log file for a Bulk API target session, select the Monitor Bulk Job Until All Batches Processed session property as well. Similarly, to generate an error log file for a Bulk API 2.0 target session, select the Monitor Bulk API 2.0 Jobs property.
Use SFDC Success File	Generates the success log files. By default, the agent does not generate the success log files. To generate a success log file for a Bulk API target session, select the Monitor Bulk Job Until All Batches Processed session property as well. Similarly, to generate a success log file for a Bulk API 2.0 target session, select the Monitor Bulk API 2.0 Jobs property.
SFDC Success File Directory	Overrides the default success file directory with the path you specify. Applicable when you select the Use SFDC Success File property. To override, specify the location in the Secure Agent machine where you want the agent to generate the success log file.
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 the external ID for the upsert operation. If you do not select this property and do not provide the external ID, the session fails.
Use this ExternalId/IdLookup field for Upserts	The exact name of the external ID or idLookup field to use for upserts. By default, the agent uses the first external ID or idLookup field in the target. Use this property when you want to use a different field for upserts.
Use SFDC Bulk API	Uses the Salesforce Bulk API to load batch files containing large amounts of data to Salesforce targets. By default, the agent uses the standard Salesforce API.
Use SFDC Bulk API 2.0	Uses the Salesforce Bulk API 2.0 to load batch files containing large amounts of data to Salesforce targets.
Monitor Bulk Job Until All Batches Processed	Monitors a Bulk API target session. When you select this property, the agent logs the status of each batch in the session log. If you do not select this property, the agent does not generate complete session statistics for the session log. Select this property along with the Use SFDC Success File or Use SFDC Error File session properties to generate the success or error logs for the session.
Monitor Bulk API 2.0 Jobs	Monitors a Salesforce Bulk API 2.0 target.

Advanced Property	Description
Override Parallel Concurrency	Instructs the Salesforce Bulk API to write batches to targets serially. By default, the Bulk operation writes batches in parallel.
Disable Bulk Success and Error File Creation	Disables the creation of success and error log files for a Bulk API and Bulk API 2.0 target session. Overrides the Use SFDC Error File and Use SFDC Success File session properties.
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. By default, the agent writes overflow data to the session error file.

Salesforce lookup objects in mapping tasks

For Salesforce lookup connections used in mapping tasks, you can configure advanced properties in the Other Parameters page of the Mapping Task wizard.

The following table describes the Salesforce lookup object properties that you can configure in a Lookup transformation:

Lookup Object Properties	Description
Connection	Name of the lookup connection.
Lookup Object	Name of the lookup object for the mapping.
Display technical names instead of labels	Displays technical names instead of business names.
Display lookup fields in alphabetical order	Displays lookup fields in alphabetical order instead of the order returned by the source system.

Custom query source type

You can use a custom query as a source object when you use a Salesforce connection.

You can use a custom query as the source when a source object is large. The custom query helps reduce the number of fields that enter the data flow. You can also create a parameter for the source type when you design your mapping so that you can define the query in the Mapping Task wizard.

To use a custom query as a source, select Query as the source type when you configure the Source transformation and then use a valid and supported SOQL to define the query.

Supported SOQL for Custom Queries

Custom queries must be valid and cannot include SOQL statements that Data Integration does not support. Data Integration supports the following SOQL statements and syntax for custom queries:

SOQL Feature	Examples
SELECT ID from Account	`SELECT Id from Account`
SELECT ID and Name from Account	`SELECT Id, Name from Account`
Fully qualified names	`SELECT Account.Id, Account.Name from Account`
Filters	`SELECT Account.Name ,Account.NumberOfEmployees, Account.SLAExpirationDate__c ,Account.utf_filter__c, Account.Datetimefld__c_FROM Account Where ((utf_filter__c LIKE '%UTFDSS %'))`
Complex filters	`SELECT Account.Name, Account.NumberOfEmployees, Account.SLAExpirationDate__c, Account.utf_filter__c, Account.Datetimefld__cFROM Account Where (((utf_filter__c LIKE '%UTFDSS %') AND Name IS NOT Null) OR AnnualRevenue > 567)`
GROUP BY	`SELECT LeadSource FROM Lead GROUP BY LeadSource`
Aggregate functions and GROUP BY	`SELECT Name, Max(CreatedDate) FROM Account GROUP BY NameLIMIT 5 SELECTCampaignId,AVG(Amount)FROMOpportunityGROUP BY CampaignId SELECT COUNT(Id) FROM Account WHERE Name LIKE 'a%'`
HAVING	`SELECT LeadSource, COUNT(Name) FROM Lead GROUP BY LeadSource HAVING COUNT(Name) > 100`
IN	`SELECT Name FROM Account WHERE BillingState IN ('California', 'New York')`
NOT IN	`SELECT Name FROM Account WHERE BillingState NOT IN ('California', 'NewYork')`
ORDER BY	`SELECT Name FROM Account ORDER BY Name DESC NULLS LAST SELECT Id, CaseNumber,Account.Id,Account.NameFROMCaseORDER BY Account.Name`
OFFSET	`SELECT Name FROM Merchandise__c WHERE Price__c > 5.0 ORDER BY Name LIMIT 100 OFFSET 10`
ROLLUP	`SELECT LeadSource, COUNT(Name) cnt FROMLeadGROUP BY ROLLUP(LeadSource)`
TYPEOF	`SELECT TYPEOF What WHEN Account THEN Phone ELSE Name END FROM EventWHERECreatedById IN (SELECTCreatedByIdFROMCase)`
Aliases	`SELECT Name n, MAX(Amount) max FROM Opportunity GROUP BY Name`
Date functions	`SELECT CALENDAR_YEAR(CreatedDate), SUM(Amount) FROM Opportunity GROUP BY CALENDAR_YEAR(CreatedDate)`

SOQL Feature	Examples
Date literals	`SELECT Id FROM Account WHERE CreatedDate = YESTERDAY SELECT Id, CaseNumber,Account.Id,Account.NameFROMCaseORDER BY Account.Name`
Multi-currency	`SELECT Id, convertCurrency(AnnualRevenue) FROM Account SELECT Id, NameFROMOpportunityWHEREAmount > USD5000`
Operators	`SELECT Id FROMAccountWHERECreatedDate > 2005-10-08T01:02:03Z`
Semi-joins	`SELECT Id, NameFROMAccountWHERE Id IN (SELECTAccountIdFROM OpportunityWHEREStageName = 'ClosedLost')`
Reference fields in semi-joins	`SELECT Id FROM Task WHERE WhoId IN (SELECT Id FROM Contact WHERE MailingCity = 'Twin Falls')`
Relationship queries in semi-joins	`SELECT Id, (SELECT Id fromOpportunityLineItems)FROM Opportunity WHERE Id IN (SELECT OpportunityId FROM OpportunityLineItem WHERE totalPrice > 10000)`
Parent-child relationship queries	`SELECT Name, (SELECT LastName FROM Contacts) FROM Account SELECT Account.Name, (SELECT Contact.LastName FROM Account.Contacts) FROM Account SELECT Name, (SELECT Name FROM Line_Items__r) FROM Merchandise__c WHERE Name LIKE 'Acme%' SELECT Id, Name,(SELECT Id, NameFROMAlldatatypeDetail__r) FROM AllDatatypes__c`
Child-parent relationship queries	`SELECT Id, FirstName__c,Mother_of_Child__r.FirstName__cFROM Daughter__c WHERE Mother_of_Child__r.LastName__c LIKE 'C%' Select Id, IsDeleted, Name, LastViewedDate, LastReferencedDate, AllDatatypes_md__c, AllDatatypes_md__r.Name from AlldatatypeDetail__c`
Relationship queries with aggregate functions	`SELECT Name, (SELECT CreatedBy.Name FROM Notes) FROM Account SELECT Amount, Id, Name, (SELECT Quantity, ListPrice, PricebookEntry.UnitPrice, PricebookEntry.Name FROM OpportunityLineItems) FROM Opportunity`

You cannot use complex relationship queries, for example:

```
SELECT task.account.name, task.who.name, task.activitydate,task.account.annualrevenue FROM task WHERE task.activitydate <= TODAY AND task.who.type = 'Contact' ORDER BY task.account.annualrevenue DESC
```

Rules and Guidelines for Salesforce Custom Queries

Use the following rules and guidelines when you use a custom query as a source:

- You can use a custom query as a source when you use the Salesforce Standard API, Salesforce Bulk API, and Bulk API 2.0.
- If you configure the SOQL filter condition under advanced properties, then the custom query overrides the SOQL filter condition.
- You can use custom objects with any data type in a custom query.
- You can use expressions with custom query fields.
- You can configure field data types for a custom query source.
- You can use a custom query as a source in a scheduled task.

Salesforce Bulk API in mappings and mapping tasks

You can use the Salesforce Bulk API and Bulk API 2.0 to read data from Salesforce sources and write data to Salesforce targets. Use the Bulk job to process large amounts of Salesforce data while generating a minimal number of API calls.

When you use the Salesforce Bulk API, Salesforce restricts the size of each batch to 10 MB of data or 10,000 records in CSV format. For example, if you want to read 10 million records, the records are split into 1000 batches and each batch contains 10000 records. If the size of the 10000 records exceeds 10 MB, the batches are split based on the size instead of the number of records.

When the mapping task creates a batch, it adds any required characters to format the data, such as adding quotation marks around text.

For Bulk API 2.0, you can process 100 million rows of data in a 24 hour period. Automatic batching of files occur at the Salesforce end without any customer intervention.

You can monitor jobs that use the Bulk API and Bulk API 2.0 to write to Salesforce targets. When you monitor a Bulk API target job, the mapping task can create success and error files for row-level information. For Bulk API, the mapping task can also load batches at the same time or serially.

Note: Use API version 50 and above for all the Salesforce Bulk 2.0 operations on the Salesforce connector.

Monitor the bulk job

When you use the Bulk API to write to Salesforce targets, you enable the task for monitoring. With monitoring enabled, the mapping task requests the status of each batch from the Salesforce service. The mapping task repeats the request every 10 seconds until all batches complete. The mapping task writes the responses from the Salesforce service to the **All Jobs** page and the session log. With monitoring enabled, the mapping task also generates a Bulk API error file.

By default, the mapping task permits monitoring for Bulk jobs. You can configure the task to run without monitoring. Without monitoring, the **All Jobs** page and the session log contain information about batch creation, but do not contain details about batch processing or accurate job statistics.

Note: The mapping task performs additional API calls when monitoring a Bulk API job. To reduce the number of API calls the mapping task makes, do not monitor the job. For more information about batch processing, use the Job and Batch IDs from the session log to access Salesforce statistics.

Success and error files for the bulk API

When you monitor a Bulk processing of a target job, the mapping task generates a Bulk API error file by default. You can configure the task to create a Bulk API success file. Success and error files are CSV files that contain row-level details provided by the Salesforce service.

The Bulk API success and error files include the job ID, batch ID, Id, success, created, and error message information. These files are applicable for both Bulk API and Bulk API 2.0 jobs.

The following table describes the location and naming convention for the Bulk API success and error files:

File Type	Location	Naming Convention
Bulk API success file	<Secure Agent installation directory>\apps\Data_Integration_Server\data\success	For Bulk API: s_mtt_<TaskID>_TimeStamp_bulk_success.csv For Bulk API 2.0: s_mtt_<TaskID>_TimeStamp_bulk_v2_success.csv
Bulk API error file	<Secure Agent installation directory>\apps\Data_Integration_Server\data\error	For Bulk API: s_mtt_<TaskID>_TimeStamp_bulk_error.csv For Bulk API 2.0: s_mtt_<TaskID>_TimeStamp_bulk_v2_error.csv

Enable serial mode

When you use the Bulk API to load data to Salesforce, you can configure the mapping or mapping task to perform a parallel load or a serial load. By default, it performs a parallel load.

In a parallel load, Salesforce writes batches to targets at the same time. Salesforce processes each batch whenever possible. In a serial load, Salesforce writes batches to targets in the order it receives them. Salesforce processes the entire content of each batch before processing the next batch.

Use a parallel load to increase performance when you do not need a specific target load order. Use a serial load when you want to preserve the target load order, such as during an upsert load.

For Bulk API 2.0, the serial mode selection is not available. The batch processing occurs in parallel load by default.

Hard deletes

When you use the Bulk API or Bulk API 2.0, you can configure the mapping to permanently delete data from Salesforce targets.

When you use a Bulk task to delete data from Salesforce targets, the mapping copies the deleted rows to the recycle bin. You can retrieve deleted data within a certain amount of time, but it also requires additional space from the hard disk.

With a hard delete, the mapping bypasses the recycle bin. You cannot recover data if you delete the data with the hard delete option.

Enable primary key chunking

When you use the Bulk API to extract data from Salesforce, you can enable primary key chunking. By default, the Bulk API does not use primary key chunking.

When you enable primary key chunking in a mapping, the Bulk API splits the data set into multiple chunks based on the record ID and creates extract queries for each chunk. The Bulk API combines the data when all the extract queries are complete.

Salesforce supports primary key chunking for custom objects and certain standard objects. Bulk API 2.0 does not support PK Chunking during a mapping since automatic file batching occurs at the Salesforce end. For more information about objects that support primary key chunking, see the Salesforce documentation.

Note: You can enable primary key chunking when your Salesforce connection uses version 32 or higher of the Salesforce API. The default chunk size is 100000.

CHAPTER 5

Replication tasks with Salesforce

You can replicate Salesforce data to a target using the replication task. You might replicate data to back up the data or perform offline reporting. You can replicate data in Salesforce objects to databases or flat files.

A replication task can replicate data from one or more Salesforce objects. When you configure the task, you can replicate all available objects through the selected connection, or you can select objects for replication by including or excluding a set of objects. You can also exclude rows and columns from the replication task. Associate a schedule with a replication task to specify when and how often the task runs.

When you replicate Salesforce sources, you can replicate all current rows in the Salesforce source. You can also replicate deleted and archived rows to preserve or analyze historical data.

When you replicate Salesforce sources to target database tables that do not yet exist, the replication task generates a non-unique index for each target table. The replication task also generates an index when you replicate Salesforce sources to target database tables when you use the **Create Target** option in the Replication Task wizard.

Replication task source properties

In a replication task, you can configure the properties of objects that you want to replicate.

The following table describes the Salesforce source properties in a replication task:

Property	Description
Source Connection	Name of the source connection.
Objects to Replicate	Source objects that you can replicate. Select All Objects, Include Objects, or Exclude Objects. **Note:** For Salesforce, only Salesforce objects that can be queried display in the Available Objects area. If an object does not appear, contact the Salesforce administrator.
If an error occurs while processing an object:	Terminates or continues processing of an object if an error occurs.
Display technical names instead of labels	Displays technical names instead of business names.
Include archived and deleted rows in the source	Includes archived and deleted rows in the source. By default, the agent returns active rows.

Optionally, configure the advanced Salesforce source properties:

Advanced Salesforce Property	Description
High Precision Calculations	Enables the processing of data with a precision up to 28 in Salesforce calculated fields. Select True to enable high precision calculations. Default is False.
Maximum Base64 Body Size	Body size for base64 encoded data. Default is 7 MB.
Use Bulk API	API used to read data from Salesforce sources. Select True to use Salesforce Bulk API to read source data. Select False to use the Salesforce standard API to read source data. Default is False.
Enable PK Chunking	Use to extract from Salesforce sources when you use the Bulk API. Enables primary key chunking to optimize performance while replicating data from large data sets. Salesforce splits the data set into a number of chunks based on the record ID, creates multiple queries to extract data, and combines the result. Salesforce supports primary key chunking for custom objects and certain standard objects. For more information about supported objects for primary key chunking, see the Salesforce documentation.
PK Chunking Size	The number of records in a chunk. Default is 100,000. The maximum value is 250,000. Applicable if you select Enable PK Chunking
PK Chunking startRow ID	The record ID from which you want to chunk the data set. By default Salesforce applies chunking from the first record. Applicable if you select Enable PK Chunking.

Replication task target properties

In a replication task, you can configure the target properties.

The following table describes the target properties in a replication task:

Property	Description
Connection	Name of the target connection. Select a target connection or create a new connection.
Target Prefix	String that prefixes the source object names to create names for the target objects in the target. **Note:** If you configure a prefix for the target table name, ensure that the prefix and corresponding Salesforce object name do not exceed the maximum number of characters allowed for a target table name. For more information, see “Table and Column Names in a Database Target” in *Tasks*.

Property	Description
Load Type	Determines all new, changed, and deleted data in the source objects since the last run, and propagates the changes to the target objects.
Delete Options	Removes or retains deleted columns and rows.
Commit size	Used for replication task for all the runs. If the value is not specified, the agent uses the default value.

Load types

The load type determines the type of operation to use when the replication task replicates data from the source to the target.

Use one of the following load types when you replicate data:

Incremental loads after initial full load

The first time the replication task runs, it performs a full load, replicating all rows of the source. For each subsequent run, the replication task performs an incremental load. In an incremental load, the replication task uses an upsert operation to replicate rows that changed since the last time the task ran. You can specify this load type when the task uses a Salesforce source and a database target.

Incremental loads after initial partial load

The replication task always performs an incremental load with this load type. The first time the replication task runs, the replication task processes rows created or modified after a specified point in time. For each subsequent run, the replication task replicates rows that changed since the last time the task ran. You can specify this load type when the task uses a Salesforce source and a database target.

Full load each run

The replication task replicates all rows of the source objects in the task during each run. You can specify this load type when the task uses a Salesforce or database source and a database or flat file target.

For information about incremental load, see the help for Salesforce Connector.

Full load

For a full load, the replication task replicates the data for all rows of the source objects in the task. Each time the task runs, the replication task truncates the target database tables or flat file and performs a full data refresh from the source.

Run a full load in the following situations:

- The replication task uses a database source.
- A Salesforce object in the replication task is configured to be nonreplicateable within Salesforce.
  If you run an incremental load on a replication task that contains nonreplicateable objects, the replication task runs a full load on the object. Contact the Salesforce administrator to get a list of replicateable Salesforce objects.

- The data type of a Salesforce field changed.
  If the replication task detects a data type change, you might need to reset the target table to create a table that matches the updated Salesforce object. Then run the replication task with full load to reload the data for all Salesforce objects included in the replication task. Alternatively, you can set the AutoAlterColumnType custom configuration property so that the target table column updates to match the Salesforce object. The AutoAlterColumnType property does not apply in certain situations, such as when the source and target data types are not compatible. For more information about the AutoAlterColumnType property, see the .

Incremental loads

You can use incremental loads when you replicate Salesforce data to a database target.

You can use the following types of incremental loads:

- Incremental load after initial full load. The first time you run the replication task, it runs a full load and replicates data from all the rows.
- Incremental load for a specific period in time. The first time you run the replication task, it performs an upsert to replicate data based on changes made to the source for a specified time period.

After the initial run, both incremental load types replicate data in the same manner.

The replication task performs an upsert operation to replicate data for new rows and the rows that have changed since the last run of the task. The time of the last run is determined by the time that the last record is replicated from Salesforce.

Each replication session occurs in a single transaction. If you use the default commit interval and errors occur, the entire transaction rolls back. You can set the commit interval to a smaller value so that if a roll back occurs, only the last batch rolls back. To optimize performance, do not use an incremental load if you schedule the replication task to run at long intervals, such as weeks or months. Instead, run a full load to avoid rollback of all data that was replicated during the schedule interval.

The replication task never truncates target tables in an incremental run. To truncate target tables, you must run a full load replication task.

When the replication task compares Salesforce source objects and target tables to find inconsistencies, it completes the following tasks:

1. Compares field names in the source and target.
2. Inserts and deletes fields in the target table to match the source.
3. Runs a query to determine if values have been inserted or updated.
4. Replicates new and changed rows.

You cannot run an incremental load with the Salesforce Bulk API. When you select the incremental load option in a task, you disable the Salesforce Bulk API advanced option.

Rules and guidelines for running incremental loads

Use the following rules and guidelines when you run a replication task as an incremental load:

- You can add a column to a target. Because the replication task reconciles field name inconsistencies before it runs the query to find changed data, changes to field names do not cause rows to be replicated. Replication occurs only if the row data changes.
  For example, the Salesforce source contains a field named New that does not exist in the target. If you run an incremental load and no value exists in row 1 for the New field, the replication task adds the New field to the target table, but it does not replicate the data in row 1. If the New field contains a value in row 1, then the replication task replicates the data in row 1.
- The incremental load might fail if the data type, precision, or scale for the source column and target column are inconsistent. The incremental load fails when the source and target columns have inconsistent data types or the source column has a higher precision or scale than the target column. To resolve this issue, you can set the AutoAlterColumnType custom configuration property so that the target table column updates to match the Salesforce object.

  Alternatively, you can reset the target to re-create the target tables to match the corresponding source objects. Next, run the replication task with full load to reload the data for all source objects included in the replication task.
- The replication task runs an incremental load on a Salesforce object that is replicateable and has the CreatedDate and SystemModstamp attributes. Each Salesforce object can be configured to be non-replicateable in Salesforce. Contact the Salesforce administrator to get a list of replicateable Salesforce objects.
  Salesforce does not track the CreatedDate or SystemModstamp dates for all objects.
- If you run a task with an incremental load multiple times, the time period between the end time of the previous run and the start time of the next run must be at least 60 seconds. The task fails if you try to run the task before the 60-second waiting period.
- You cannot run a task with an incremental load when the target is a flat file.

Database target reset

Reset a relational target table in a replication task to drop all of the target tables in the task.

You might need to reset a relational target for the following reasons:

- If the data type, precision, or scale of a Salesforce source field changed and you run the replication task as an incremental load, the task might fail. This is because of a mismatch between the data types of the source field and the column of the relational target table. The task might also fail because the target table column might not be able to store all values from the Salesforce field. You can reset the target table to synchronize the data type, precision, or scale of the target table column with the Salesforce field. If the precision or scale decreases, the replication task succeeds and the target table columns remain unchanged.
- You delete a field in a Salesforce source and the replication task writes to a Microsoft SQL Server relational target.
  If you run a replication task that writes to a Microsoft SQL Server target and the source is missing a field or column, the replication task fails. To run the replication task successfully, reset the target table to re-create the target based on the latest source definition, and then run the replication task.

Handling source and target mismatch

When you perform an incremental run with a Salesforce source and a database target, the task might fail if the source and target metadata do not match.

The task can fail because the data type in the Salesforce field and data type for the database target column do not match. The task can also fail if the precision or scale for a Salesforce field increases and the database column might not be able to store the value.

To adjust the target column for changes to source metadata, you can use the AutoAlterColumnType custom configuration property. When you use this property, the replication task modifies the target column metadata to match the source. Set the AutoAlterColumnType custom configuration property for the Secure Agent that runs the replication task.

You cannot use the AutoAlterColumnType property in all situations. If the AutoAlterColumnType property does not resolve a mismatch, you might need to reset the target and reload.

The following table includes a few examples of metadata differences and possible approaches to take:

Source and Target Metadata Mismatch	Action
Increase in precision	Use AutoAlterColumnType.
Decrease in precision	Reset target and reload.
Change in data type	Reset target and reload.
Additional column	No action required. All load types can resolve this issue.
Deleted column	No action required. All load types can resolve this issue.

Setting the AutoAlterColumnType property

Set the AutoAlterColumnType custom configuration property so the database target column adjusts when the data type, precision, or scale of a Salesforce source field changes. Set this property for the Secure Agent that runs the replication task.

1. Click **Configure** > **Runtime Environments**.
2. Click on the Secure Agent that runs the replication task and select **Edit Secure Agent**.
3. Under **Custom Configuration Details**, click the **Add** icon and enter the following values:

Field	Value
Type	Select **Tomcat**.
Sub-type	Select **INFO**.
Name	The name of the custom property. Enter `AutoAlterColumnType`.
Value	Enter `yes` to turn on this property.

4. Click **OK** to save your changes.

High precision calculations

You can enable high precision calculations in a task to process high precision data in Salesforce calculated fields. When you enable high precision calculations, the replication task can read data with a precision up to 28 in Salesforce calculated fields and write the data to the target.

Salesforce Base64 encoded body size

For Salesforce sources, you can configure the body size for base64 encoded data.

Configure the base64 body size when a Salesforce source includes base64 encoded data that you want to process in a replication task. By default, the body size for base64 encoded data is 7 MB. You can increase the size as necessary.

Note: If you configure a synchronization task to process this data after replication, use the Edit Types option to increase the precision of the data type for the base-64 encoded data.

Salesforce API

You can use the Salesforce standard or Bulk API to process Salesforce data in a replication task.

Use the standard API to process a normal amount of Salesforce data with standard reporting on the results of the standard API load.

Use the Bulk API to process large amounts of Salesforce data while generating a minimal number of API calls.

With the Salesforce Bulk API, each batch of data can contain up to 10,000 rows or one million characters of data in CSV format. When the replication task creates a batch, it adds any required characters to properly format the data, such as adding quotation marks around text.

When you use the Bulk API, enable primary key chunking to increase performance when you extract data from large tables. When you enable primary key chunking, the Bulk API splits the data set into chunks based on the record ID and creates extract queries for each chunk. The Bulk API combines the data when all the extract queries are complete. Salesforce supports primary key chunking for custom objects and certain standard objects.

Rules and guidelines for configuring replication tasks

Consider the following rules and guidelines for configuring replication tasks:

- The names of source tables and fields can contain at most 79 characters.
- You cannot configure a replication task with the same source and target objects. If the source and target connections are the same, you must enter a target prefix to distinguish the source and target objects.
- You cannot replicate data to a Salesforce target.

- You cannot configure multiple replication tasks to replicate the same source object to the same target object. For example, you cannot configure two replication tasks to write Salesforce Account data to the SF_ACCOUNT Oracle database table.
- You cannot simultaneously run multiple replication tasks that write to the same target table.
- When you configure a replication task to run on a schedule, include a repeat frequency to replicate the data at regular intervals.
- The index is generated based on the Salesforce ID field. Indexes are not generated for Salesforce sources that do not include a Salesforce ID field.
- You cannot simultaneously run replication tasks that replicate the same Salesforce object. If you simultaneously run two replication tasks that replicate the same object, the replication task that starts first obtains a lock on the Salesforce object, and the other replication task fails with the following error:

  ```
  replication task failed to run. replication task name> is currently replicating the
  same objects.
  ```

  If you configured the replication tasks to run on schedules, schedule the replication tasks to run at different times. If you run the replication tasks manually, wait until one completes before your run the other.
- You can load data from Salesforce fields of any datatype, except binaries, into a flat file target.
- There must be a time difference of at least 60 seconds between two consecutive runs of a replication task with incremental load.
- To adjust the database target column when the data type, precision, or scale of a Salesforce source field changes, you can set the AutoAlterColumnType custom configuration property. Set this property for the Secure Agent that runs the replication task. In the Secure Agent properties, select the type as Tomcat and the subtype as INFO. Then, enter AutoAlterColumnType as the name and set its value to yes to turn on this property.

CHAPTER 6

Masking tasks with Salesforce

The masking task masks the sensitive fields in source data with realistic test data for nonproduction environments. You can choose to create a subset of the sensitive source data with object relationships reconciled.

When you create a masking task, you can use standard objects to add multiple source objects. The masking task uses external IDs, custom fields, or unique field lookup to reconcile relationships between the target parent-child objects. When you configure targets, you can specify external IDs to write data into the targets.

When you select multiple source objects in a masking task, you select a primary object and add the required related objects individually. You can use junction objects to store the relationship details between two Salesforce objects.

When you run a masking task, the task can populate the source owner name of the User object in the target instead of the target connection name.

Relationship reconciliation strategies

In Salesforce, a record ID uniquely identifies the records and associates a record with the other. Salesforce uses the record ID to reconcile the parent-child relationships.

When you run a masking task, the task reconciles relationships through an external ID, a custom field, or a unique field, and writes the data to the target. Salesforce recommends to create and use external IDs instead of custom field lookup to insert or upsert the target data. If Salesforce cannot create an external ID, the masking task creates a custom field in Salesforce to perform lookup on the target. If Salesforce cannot create an external ID or a custom field, the masking task creates a unique field to perform lookup on the source and the target. The external ID field or the custom field lookup uniquely identifies the parent-child relationship in a record when you insert or upsert data in a masking task.

External ID field

A Salesforce external ID field contains the External ID attribute with unique record identifiers from a system outside Salesforce.

The Data Masking task uses external IDs to identify the parent-child relationships objects in the target database. Salesforce recommends to create and use external IDs instead of custom field lookup or unique field lookup to insert or upsert the target data. When you create an external ID from the Target tab, the Data Masking task appends DMASK_ to the name of the external ID. The Target tab shows the external IDs that you create in the Data Masking task.

The Data Masking task creates an additional field for the external ID in the target at the design time. If an external ID exists for the object, the task uses the same external ID. You can either retain or delete the

external IDs after you run the task. You can retain the external ID fields if you want to perform another upsert operation.

If you cannot create an external ID field for an object, you can select a unique field or an idlookup field for the object from the Target tab when you perform an upsert operation. For example, you can select a unique field or an idlookup field for the RecordType object.

Custom field lookup

The masking task performs a lookup-based reconciliation on the parent target object to get a parent record ID.

When the external ID is not present, the masking task creates a custom field to perform a lookup in the Salesforce target with the same name. A custom field lookup requires one lookup operation on the target.

The masking task performs a custom field lookup in the following situations:

- The external ID limit exceeds in the target object.
- The number of unique fields in the target exceeds the limit.

Unique field lookup

The masking task uses a unique field for the objects on which you cannot create an external ID or a custom field. This reconciliation strategy is for standard objects that contain unique fields.

If there are unique fields present in the object, the task performs the lookup operation based on the unique field. A unique field lookup requires one lookup operation on the source and the target.

If you cannot create an external ID field for an object, you can select a unique field or an idlookup field for the object from the Target tab when you perform an upsert operation. For example, you can select a unique field or an idlookup field for the RecordType object.

If you cannot create an external ID field on an object and if there is no unique field present for the object, you cannot perform an insert operation. For example, you cannot perform an insert operation on the OpportunityContactRole object.

The unique field lookup strategy is applicable to the following standard objects:

- AdditionalNumber
- Announcement
- ApexClass
- ApexComponent
- ApexPage
- ApexTrigger
- Attachment
- AuthProvider
- BrandTemplate
- BusinessHours
- BusinessProcess
- CallCenter
- CollaborationGroup
- ContentDistribution

- CorsWhitelistEntry
- Document
- EmailServicesAddress
- EmailServicesFunction
- EmailTemplate
- EntitlementContact
- EntitlementTemplate
- Folder
- Group
- Holiday
- LiveChatTranscriptEvent
- LiveChatTranscriptSkill
- LiveChatUserConfigProfile
- LiveChatUserConfigUser
- LiveChatVisitor
- MailmergeTemplate
- MilestoneType
- NetworkActivityAudit
- Note
- PresenceUserConfigProfile
- PresenceUserConfigUser
- QuestionReportAbuse
- QuestionSubscription
- QuoteDocument
- RecordType
- ReplyReportAbuse
- SelfServiceUser
- RecordType
- StreamingChannel
- Topic
- TopicAssignment
- User
- UserProvAccount
- UserProvAccountStaging
- UserProvMockTarget
- UserProvisioningLog
- UserRole
- WebLink

Junction objects

A junction object is a Salesforce object that contains many-to-many relationships between two related objects.

The relationship details stored within a junction object form a junction relationship. In a many-to-many relationship, each record in an object links to multiple records in another object. A junction object stores all the relationships between the two objects. For example, CaseSolution is a junction object that stores many-to-many relations between the Case object and the Solution object. The relationship between the Case object and the Solution object is the junction relationship.

You can create a data subset from a junction object. You can insert data into a junction object. You cannot upsert data into the junction object.

Target owner name

In Salesforce, the masking task can add the source owner name of the objects in the target instead of the target connection user name. The target must contain a user with the same alias.

When you select multiple sources, the User object and the other related objects are added to the list of sources. The User object reconciles the source owner name based on the Alias field in Salesforce. If the target contains the same owner name as the source, the masking task adds the source owner name in the target. If the target does not contain the same owner name as the source, the masking task populates the default target connection name.

If multiple users have the same alias name, the masking task choose a random user to reconcile relationship with the other object.

Salesforce bulk API limits

Salesforce limits the amount of data that you can read or write through the Salesforce Bulk API.

To improve the performance and reduce the number of API requests for large data sets, you use Salesforce Bulk API.

Batch Limit

You can submit up to 5,000 batches in every 24 hours. You can perform query, write, and delete operations.

Bulk API Writer

Default size of the Bulk API Writer is 10,000 rows for a batch. The task can process up to 5000*10000 which is 50 million records a day. Edit the Secure Agent properties to configure the batch size of the Bulk API Writer.

Bulk API Reader

With Salesforce Bulk API Reader, you can retrieve up to 15 GB data from a query. If the query exceeds 15 GB, the masking task fails. You can manually calculate whether the amount of data that you want to process exceeds the Salesforce limitations.

To calculate the amount of data you can query through the Salesforce Bulk API Reader, you can use the following formula:

```
Sum (the number of bytes in the fields that you want to mask) * (the number of rows in
the query results) < 15 GB
```

In the Salesforce application, view the size of the fields of the objects and add the number of bytes of the fields that you want to mask. Find the total number of rows in the query results for the Salesforce object and

apply the formula. If the result is within the 15 GB limit, the task succeeds. If the result is greater than 15 GB, the task fails.

Salesforce Bulk API Limits Example

The Account object contains 100 fields and you want to mask 10 fields of sensitive data. The sum of the size of all the 10 fields that you want to mask is 100 bytes. The total number of rows for the Account object is 500,000.

Use the following formula:

Total query size = 100 * 500,000 = 0.05 GB

The result 0.05 GB is within the 15 GB data limit and the application can process the task successfully.

The following table contains some more examples of total query size calculations for the Salesforce objects:

Objects	Number of Rows in Query Results	Sum of the Size of the Fields to be Masked	Total Query Size
Account	500,000	1,000	0.5 GB
Contact	6,000,000	2,000	12 GB
Lead	8,500,000	2,000	15.8 GB

The total query size for the Account and Contact objects are within the 15 GB data limit and the application processes the task successfully. The total query size for the Lead object is greater than the 15 GB data limit and the application cannot process the data. To reduce the query for the Lead object, you can use horizontal or vertical partitioning and create multiple masking tasks.

In horizontal partitioning, you can split the number of rows in a Salesforce object. Specify a condition in the data filter criteria to reduce the total number of rows in the query results. You can run two masking tasks, one with closed Leads and the other with open Leads.

In vertical partitioning, you can split the number of fields in a Salesforce object. For the Lead object, you can create 2 masking tasks, 1 task to mask 10 fields with 1300 bytes of data, and other task to mask 3 fields with 700 bytes of data.

Salesforce limitations in masking

Salesforce has the following limitations for masking tasks:

- You cannot apply filters on the TextArea and TextEncrypted field types.
- Salesforce has limit of 20,000 characters on the size of a query. You can configure source or target profiles to hide the fields to reduce the number of fields in a object.
- You cannot perform an upsert operation if the Salesforce user account does not have creatable or updatable permissions on the external ID.
- You cannot use the User, UserLicense, Profile, Community, and Group objects in a masking task.

Special handling of standard objects

If you cannot create an external ID field in a child object or a junction object, create an external ID with the insert operation. Change the task operation from insert to upsert before you run the task.

The following standard objects require special handling:

Pricebook2

Every organization contains a standard Pricebook object that you cannot alter. If the source contains the Pricebook2 object, you must copy the Pricebook2 object ID to the corresponding external ID field in the target after the task creates the external ID in the target and before you run the task.

RecordTypes

You can select the RecordTypes object only when you enable them at profile level for individual objects. Before you run the task, enable the required RecordTypes in the target profile.

AssetRecord

To create the AssetRecord object, you must add the Account object or the Contact object to the Asset object in a task or else the task fails. Since these relationships are not mandatory, there are no errors or warnings in the validation report.

PersonAccounts

You cannot edit the AccountName field directly in the PersonAccounts object. Salesforce combines the FirstName and LastName fields in the appropriate order according to the language settings that you configure and derives the AccountName field. You can view and edit the AccountName field. You cannot alter the field level security to make it a read-only field. To avoid writing to the AccountName field, apply nullification masking on the AccountName field with the PersonAccounts object.

Contracts

You cannot insert and activate the Contracts object in a single step. Insert the Contracts object with Draft status and then update with Activated status. If you attempt to insert a Contract with Activated status, you get the Invalid Status error.

Idea

Use the Idea object only for inplace masking. The Idea object has a mandatory parent relationship with the Community object. But the Community object contains read-only fields. When the target is different from the source, the task fails to insert records into the Community object.

OpportunityLineItem

In the OpportunityLineItem object, you can view and update the Unit Price and Total Price fields. When you insert a record, specify only one of the fields. Else, the task fails and generates an error. By default, both the Unit Price and Total Price fields are specified. To avoid errors, apply nullification masking on one of the fields.

User

The task can reconcile relationships for the User object in two ways. By default, the task uses the Alias field for reconciliation if the corresponding User relationships are selected in the Source page.

The other way to reconcile the relationship is to apply custom substitution masking on the Owner ID field for an object. In this case, you do not need to select User relationship in the Source page because the task overrides the default alias reconciliation. To apply custom substitution masking, perform the following steps:

1. Create a dictionary file with the target and the corresponding source user IDs.

2. Copy the file into the dictionary file location of the Secure Agent installation directory.
3. Create a flat file connection to the dictionary file directory.
4. Use the dictionary file to configure custom substitution masking on the Owner ID field.
5. Provide a target user ID in the lookup error constant if the target does not contain the Owner ID same as the source.

Advanced Salesforce options

You can view and configure advanced options for Salesforce objects.

Configure the following advanced option for Salesforce objects:

Disable Rules and Triggers

You can disable all the target Salesforce rules and triggers before you run a task. Disable the rules and triggers to increase the speed of the target data load.

A Salesforce target might contain triggers, validation rules, and workflow rules that are configured for the Salesforce objects. After you apply masking rules and run the task, the masked data might not satisfy the rules or triggers that are defined in the target. The task cannot insert the records in to the target and results in error rows. To ensure that there are no such errors, you can disable all the target Salesforce rules and triggers and run the task. After the task is complete, the task restores the previous state of all the Salesforce triggers and rules.

If the Salesforce managed packages contain rules and triggers, you cannot disable the rules and triggers in a masking task.

Use the Bulk API

Use the Salesforce Bulk API to load batch files containing large amounts of data to Salesforce targets. By default, the a pplication uses the Salesforce standard API. With the Salesforce standard API, each batch of data can contain up to 200 rows. With the Salesforce Bulk API, each batch of data can contain up to 10000 rows. When you select this option, the application monitors the Salesforce Bulk API job.

Enable Primary Key Chunking

Enable primary key chunking to increase performance when you extract data from large tables. Use primary key chunking if the API version of the connection is 32.0 and higher. When you use the Bulk API to extract data from Salesforce, you can enable primary key chunking. By default, the Bulk API does not use primary key chunking. Salesforce has a limitation of 15 GB on the size of data that you can query in one bulk query. Salesforce allows 15 retries for one query and then cancels it.

When you enable primary key chunking, the Bulk API splits the data set into multiple chunks based on the record ID and creates extract queries for each chunk. The Bulk API combines the data when all the extract queries are complete.

Enable Primary Key Chunking Sze

The number of records in a chunk. Applicable only if you select **Enable Primary Key Chunking**.

You can specify the chunk size. Default is 100,000. The maximum value is 250,000.

CHAPTER 7

Common configuration

You can perform the following configuration procedures that are common to multiple types of tasks :

- Creating a custom external IDs.
- Searching objects.
- Displaying business name.
- Creating data filter.
- Starting tasks with Salesforce outbound messages.

Salesforce targets and IDs for related objects

Data Integration identifies records of a Salesforce object based on one of the following types of IDs:

- Salesforce ID
  Salesforce generates an ID for each new record in a Salesforce object.
- External ID
  You can create a custom external ID field in the Salesforce object to identify records in the object. You might create an external ID to use the ID generated from a third-party application to identify records in the Salesforce object. You can use one or more external IDs to uniquely identify records in each Salesforce object.

If you create a task that writes to a Salesforce target, the source must provide either the Salesforce IDs or the external IDs for the records in the Salesforce target object and applicable related objects. To configure external IDs for related Salesforce objects, use **Related Objects** to select the external ID for each applicable related object. A related object is an object that is related to another object based on a relationship defined in Salesforce. The task uses the Salesforce ID or external ID to update changes to related objects.

If the source in a task contains external IDs for Salesforce objects, you must specify the external IDs for all related objects when you create the Salesforce target for the task. If you do not specify the external ID, Data Integration requires the Salesforce ID to identify records in each related object.

For more information about creating and using Salesforce external IDs, see the Data Integration Community article, "Using External IDs and Related Objects in Informatica Cloud".

Object search and selection

In Salesforce connections, you can search for the object or objects that you want to use. You can search for objects in mappings and task wizards.

You can select the following search parameters when you use a Salesforce connection:

- Name
- Label

Display business names

Salesforce connection displays business names for field names in the listed task types. You can configure tasks to display technical names instead of business names with the **Display technical field names instead of labels** option.

In Salesforce connection, the following task types display business names for field names:

- Synchronization
- Mapping

When you use a Salesforce connection, synchronization, and mapping tasks also display business names for objects when available. Other task types display technical names for objects.

Data filters

When you use a Salesforce connection, you can create the following type of data filters for any type of task:

- Simple
  Create one or more simple data filters. When you create multiple simple data filters, the associated task creates an AND operator between the filters and loads rows that apply to all simple data filters. For example, you load rows from the Account Salesforce object to a database table. However, you want to load only accounts that have greater than or equal to $100,000 in annual revenue and that have more than 500 employees. You configure the following simple data filters:

Field	Operator	Field Value
AnnualRevenue	greater than or equals	100000
NumberOfEmployees	greater than	500

- Advanced
  Create an advanced data filter to create complex expressions that use AND, OR, or nested conditions. You enter one expression that contains all filters. The expression that you enter becomes the WHERE clause in the query used to retrieve records from the source.

For example, you load rows from the Account Salesforce object to a database table. However, you want to load records where the billing state is California or New York and the annual revenue is greater than or equal to $100,000. You configure the following advanced filter expression:

```
(BillingState = 'CA' OR BillingState = 'NY') AND (AnnualRevenue >= 100000)
```

When you create a data filter on a Salesforce object, the corresponding task generates a SOQL query with a WHERE clause. The WHERE clause represents the data filter. The SOQL query must be less than 20,000 characters. If the query exceeds the character limit, the following error appears:

```
Salesforce SOQL limit of 5000 characters has been exceeded for the object:
<Salesforce object>.
Please exclude more fields or decrease the filters.
```

You can create a set of data filters for each object included in a synchronization task. Each set of data filters act independently of the other sets. You can use the following data filter variables in simple and advanced data filter conditions:

- $LastRunDate
- $LastRunTime

Note: Consider time zone differences when comparing dates across time zones. The date and time of the $LastRunDate and $LastRunTime variables are based on the time zone set in the Data Integration Salesforce application. The date and time of the actual job is based on the GMT time zone for Salesforce sources.

Rules and guidelines for data filters

Consider the following rules and guidelines for data filters for Salesforce Connector tasks:

- Salesforce fields of LongTextArea datatype do not show up in the list of fields of a simple data filter.
- When you include a Salesforce field of URL datatype, exclude “http://” from the value. For example, if the value is http://www.informatica.com, enter www.informatica.com.
- When you include a Salesforce field with the Phone datatype, enter a value with the following syntax (XXX) XXX-XXXX. For example, enter (555) 555-1212. If you provide an incorrect syntax, the application ignores the filter.
- When you include a Salesforce ID field in a filter, enter the exact ID value. If you enter a dummy ID value, the SOQL query fails.
- When you write Salesforce data to a database target in a synchronization task, verify that the Salesforce data uses the following required formats for date and time datatypes: Date (yyyy-MM-dd) and DateTime (yyyy-MM-dd HH:mm:ss).
  If a record contains the date and time in a different format, the application rejects the row. If the Salesforce source contains a date and no time for the datetime datatype, the application appends ‘00:00:00' at the end of the date value to ensure the value is in the required format.

  When you write to a database target, the application converts the Salesforce date and datetime data to the correct format expected by the database.

Starting tasks with Salesforce outbound messages

You can configure synchronization, data transfer or mapping tasks to start when they receive a Salesforce outbound message.

The task must include a single Salesforce source and cannot include row limits or data filters.

When you configure a task to start when it receives Salesforce messages, the task wizard generates an endpoint URL. In Salesforce, use the endpoint URL to configure a workflow rule so that Salesforce sends messages to Data Integration.

Note: When you use a serverless runtime environment, you cannot configure tasks to start with Salesforce outbound messages.

You can determine when the task times out. The timeout is the length of time that the task can be inactive but still available to receive messages. If a message is received during this period, the time resets. Use Custom Timeout or No Timeout options to increase the timeout length so the task continues to run and is ready to process messages. The default timeout is 60 seconds. You can specify the time in seconds for Custom Timeout. No Timeout sets the indefinite timeout.

To configure a task to run when Salesforce sends an outbound message, perform the following steps:

1. In Data Integration, select a single Salesforce source for the task in the wizard.
   Configure the task to run in real time upon receiving an outbound message from Salesforce, and copy the endpoint URL.

   Optionally, select a timeout option for the task.
2. In Salesforce, create a workflow rule and add an outbound message for the workflow action.
   Configure the Salesforce outbound message using the endpoint URL from the task wizard, and select the fields to send to the task.

   Note: If you did not copy the URL from the task wizard, you can find the endpoint URL on the task details page.

   Activate the task.

For more information about configuring outbound messages in Salesforce, see the article, "Real Time Data Synchronization Through Salesforce Outbound Messages".

CHAPTER 8

Troubleshooting

Use the following sections to troubleshoot errors in Salesforce Connector. For a list of common error messages and possible solutions, see the article, "Troubleshooting: Common Error Messages".

Troubleshooting a Salesforce connection

The solution to the following situation might help you troubleshoot Salesforce connections:

The connection fails to connect to a Salesforce account.

You may have to enter a Salesforce security token in the Salesforce connection details. If the security token is required and the Security Token field in the Salesforce connection is empty or invalid, the following error message appears when you test or create the connection:

```
The login to Salesforce.com failed with the following message -
LOGIN_MUST_USE_SECURITY_TOKEN:
```

Go to the Salesforce web site to obtain the security token. To avoid adding the security token to connection details, you can also add Data Integration IP addresses to Trusted IP Ranges in your Salesforce account. For more information, see Security Tokens and Trusted IP Ranges in the Connect to Snowflake topic. You can find additional information with in the Informatica How-To Library article: Activating IP Addresses for Salesforce Connections.

Troubleshooting a Salesforce synchronization task

The solution to the following situations might help you troubleshoot Salesforce synchronization task.

The synchronization task fails.

You tried to run a task that writes data from a flat file to a Salesforce object, but one of the external ID fields in the flat file is not found in the related Salesforce object. When this occurs, the task fails with following error:

```
[Salesforce object name] : Error received from salesforce.com. Fields []. Status
code [INVALID_FIELD].
Message [Foreign key external ID: not found for field <field name> in entity <source
object>].
```

To resolve, verify that the external ID values exist in Salesforce.

A synchronization task that is configured to run in real time upon receipt of an outbound message fails with the following message:

```
The buffer [] returned from outbound message queue [] is corrupt.
```

The following table describes the cases where this can occur and possible actions to take:

Case	Suggested Action
The fields sent by the outbound message do not match the source fields used in the task.	Update the outbound message or update the task so that the fields sent by the outbound message match the source fields used in the task.
The source metadata, such as the precision or data type of a field, has changed since the task was configured.	Update the task. To refresh the source metadata in the task, on the Field Mapping page, click Refresh Fields.

If errors persist, you can configure the following custom configuration properties for the runtime environment to resolve this problem:

- Type: Tomcat, Name: InfaAgent.MaxTotalBytesQueued, Value: <large number in MB>
- Type: Tomcat JRE, Name: INFA_MEMORY, Value: -Xmx<large number>
- Type: Tomcat, Name: InfaAgent.OutboundMessageIdDuplicateTimePeriod, Value: <large number in milliseconds>

For example:

- Type: Tomcat, Name: InfaAgent.MaxTotalBytesQueued, Value:256
- Type: Tomcat JRE, Name: INFA_MEMORY, Value: -Xmx1g
- Type: Tomcat, Name: InfaAgent.OutboundMessageIdDuplicateTimePeriod, Value: 60000

For more information about configuring a task to begin when receiving outbound messages, see "Starting tasks with Salesforce outbound messages" on page 57.

Troubleshooting a mapping task

The error file for the mapping task is empty.

When you run a mapping task and if the task fails, the error file that you download from My Jobs page is empty.

You can access the error file from the following directory:

```
Informatica Cloud Secure Agent/apps/Data_Integration_Server/data/error
```

The tasks fail with `[ERROR] java.lang.OutOfMemoryError: Java heap space` **for Salesforce Bulk API 2.0.**

Increase the java heap size on the Secure Agent machine and run the tasks again. For more information, see the following knowledge base article: KB 336913

Troubleshooting a replication task

The solution to the following situation might help you troubleshoot Salesforce replication task:

I cannot find the Salesforce UserRecordAccess or UserProfileFeed objects in the list of available objects to replicate. Also, I cannot run an existing task that includes those objects.

The replication task does not replicate the following Salesforce objects:

- ContentDocumentLink
- FeedComment
- FeedItem
- UserProfileFeed
- UserRecordAccess
- Vote

If you have a replication task from a previous version of Data Integration that includes these objects, the replication task does not run the task. To enable the task to run, remove the objects from the task.

I ran a replication task, but I did not get results in the expected time period.

Salesforce servers have an outage or are experiencing heavy traffic.

Why did my replication task fail?

A replication task might fail for the following reason:

- You run an incremental load after the data type, precision, or scale of a Salesforce field changed. To replicate Salesforce data after these changes, configure the replication task to perform a full load.

A task that includes Salesforce calculated fields with high precision data fails with truncation errors.

You can process data in calculated fields with a precision up to 28 by enabling high precision calculations for the task.

Troubleshooting masking task errors

When you perform a masking task, the following errors can occur:

Either the target does not contain the Account object, or there is a problem connecting to the source or target. Verify the source connection, target connection, and source and target objects. If the problem persists, contact Informatica Global Customer Support.

This error occurs when the masking task is not able to connect to the target or there are some objects are missing in the target connection.

REQUEST_LIMIT_EXCEEDED: Total Requests Limit exceeded.

This error occurs when you exceed the API usage limit for the day.

Error loading into target [Object_Name] : Error received from salesforce.com. Fields []. Status code [STORAGE_LIMIT_EXCEEDED]. Message [storage limit exceeded].

This error occurs when the target object is full or exceeds the limit.

Reattempt the Salesforce request [handleBulkApiError] due to the error [Invalid session id].

This error occurs when there is a temporary problem with Salesforce.com or the network. To check whether there are any known server issues, you can refer to http://trust.salesforce.com.

Error loading into target [RecordType] : Error received from salesforce.com. Fields []. Status code [DUPLICATE_VALUE]. Message [duplicate value found: <unknown> duplicates value on record with id: <unknown>].

This error occurs when a Record Type with the same name exists in the target.

The [Object_Name] object does not contain any fields that you can update. Remove the object and try again.

This error occurs when you cannot update an object or add a record. You must remove the object and run the task. This error can also occur when you have read-only access to the object. Verify that you have the right permissions.

Error : Failed to create the external ID field in the target [Object_Name](Error : External ID creation for this object is not currently supported. Select another object. If the problem persists, contact Informatica Global Customer Support.)

This error occurs when the application fails to create an external ID in the target. To upsert the data into target object, you need an external ID field. If the application does not allow to create the external ID fields, then you cannot upsert data into the target objects.

The selected object repeatedly calls one or more mandatory objects that are not supported. Remove the object and run the task.

This error occurs when the Salesforce objects defined in the composite objects do not exist in the source organization.

Cloud Masking mapgen failed. Contact Informatica Global Customer Support.

com.informatica.ilm.tdms.mapgen.exceptions.CyclicRelationException: Cyclic schemas are not supported on Salesforce.

This error occurs due to cyclic relationships within the object sets. Remove the objects that contain such cyclic relationships.

INSUFFICIENT_ACCESS_ON_CROSS_REFERENCE_ENTITY

This error occurs when the target does not contain all the related objects and there are no sufficient permissions to write on to the target.

Masking page is empty for inplace and instream masking.

This error occurs when you have read-only access to source for inplace masking or when you have read-only access to target for instream masking. When you create a masking task, the masking page is blank and you cannot view any fields.

SFDC_31102 [FATAL] Query failed. [MALFORMED_QUERY: SOQL statements cannot be longer than 20000 characters.]

This error occurs when the SOQL exceeds 20000 characters. You must contact Salesforce to increase the SOQL character limit.

Error loading into target [OpportunityLineItem]: Error received from salesforce.com. Fields [UnitPrice]. Status code [FIELD_INTEGRITY_EXCEPTION]. Message [field integrity exception: UnitPrice (only one of unit price or total price may be specified)].

This issue occurs because in the OpportunityLineItem object, both the UnitPrice and TotalPrice fields are visible and can be updated. The task fails if you insert records into both the fields. To resolve the issue, apply nullification masking on the UnitPrice field or the TotalPrice field while you insert a record into the OpportunityLineItem object.

Error loading into target [OpportunityContactRole] : Error received from salesforce.com. Fields []. Status code [INVALID_FIELD]. Message [Foreign key external ID: 003o000000a67jyaay not found for field DMASK_EXTERNAL_ID__c in entity Contact].

This issue occurs because you can neither create an external ID nor select a unique field for the OpportunityContactRole object when you perform an insert operation. Perform an upsert operation and select the idlookup field for the OpportunityContactRole object to avoid the error.

Troubleshooting a bulk job

If you use Salesforce Bulk API 2.0 to read and write data, you might experience slower performance than when you read and write data using Salesforce Bulk API.

To know the behavior of Salesforce Bulk API 2.0, see the Salesforce documentation.

Unfinished batches in a bulk job due to Salesforce session timeout

When you run a task in bulk mode and the Salesforce session times out, there may be unfinished batches in the job.

To resolve this issue, you can enable a rerun for unfinished batches and limit the batch processing time.

Perform the following steps to enable rerun for unfinished batches:

1. Log in to Informatica Cloud and select the **Administrator** service.
2. Click **Runtime Environments**.
3. Select the runtime environment that contains the Secure Agent that you have used to run the task.
4. Click **Edit**.
5. Under **System Configuration Details**, configure the following fields:
   - Service: Data Integration Server
   - Type: DTM

   The list of DTM services is displayed.
6. Click **Edit Agent Configuration** for any one of the JVM Options. For Example, JVMOption1.
7. Set the value of the JVM option to `-DSFDCDisableBatchRetry=false`.
   - True - Unfinished batches in the job are not retried.
   - False - Unfinished batches in the job are retried the standard number of times, which is 15 for bulk queries and 10 for bulk uploads.
8. Click **Save**.

After you save the changes, the Secure Agent restarts automatically.

APPENDIX A

Data type reference

Data Integration uses the following data types in mappings, synchronization tasks, and mapping tasks with Salesforce:

- Salesforce Native Data Types
  Salesforce data types appear in the Source and Target transformations when you choose to edit metadata for the fields.
- Transformation Data Types
  Transformation data types are the set of data types that appear in the transformations. Transformation data types are internal data types based on ANSI SQL-92 generic data types, which Data Integration uses to move data across platforms. Transformation data types appear in all transformations in Mappings, synchronization tasks, and mapping tasks.

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

Salesforce datatypes and transformation datatypes

The following table lists the Salesforce datatypes that Data Integration supports and the corresponding transformation datatypes:

Salesforce Datatype	Transformation Datatype	Description
AnyType	String	Polymorphic data type that returns string, picklist, reference, boolean, currency, integer, double, percent, ID, date, datetime, URL, or email data.
Base64	String	Base64 encoded binary data.
Boolean	Integer	Boolean (true/false) values.
Combobox	String	Enumerated values.
Currency	Decimal	Currency values.
DataCategoryGroupReference	String	Types of category groups and unique category names.
Date	Date/Time	Date values.

Salesforce Datatype	Transformation Datatype	Description
DateTime	Date/Time	Date and time values.
Double	Decimal	Double values.
Email	String	Email addresses.
Encrypted String	String	Encrypted text fields contain any combination of letters, numbers, or symbols stored in encrypted form.
ID	String	Primary key field for a Salesforce object.
Int	Integer	Fields of this type contain numbers with no fraction portion.
JunctionIdList	String	A string array of referenced ID values that represents the many-to-many relationship of an underlying junction entity. Query and manipulate the string array to query and manipulate the underlying junction entities in a single API call. **Note:** This field type is available from V34.0 API and above.
Location	Decimal	A compound data type that contains latitude and longitude values (Double) for geolocation field.
Master record	String	ID of the merged record.
Multipicklist	String	Multiple-selection picklists, which provide a set of enumerated values that you can select multiple values from.
Percent	Decimal	Percentage values.
Phone	String	Phone numbers.
Picklist	String	Single-selection picklists, which provide a set of enumerated values that you can select one value from.
Reference	String	Cross-references to another Salesforce object.
String	String	Character strings.
Textarea	String	String that appears as a multiple-line text field.
Time	Date/Time	Time values.
URL	String	URL values.

Rules and guidelines for datatypes

Consider the following rules and guidelines for Salesforce Time datatype and Transformation datatypes:

- Data Integration reads the Salesforce Time datatype as Date/Time datatype.
- If you write a Date/Time value with precision to the millisecond, to Salesforce, Data Integration truncates the millisecond.

- If you want to read a Time value with precision to the millisecond, you must read the Time datatype as AnyType datatype or String datatype.
- You must create a mapping task to perform a lookup operation on the Time datatype.
- If you want to use a saved query, you must change the Time value to AnyType datatype or String datatype.
- Use advanced filter instead of simple filter for Time datatype.

INDEX

A

AutoAlterColumnType property 45

C

Cloud Application Integration community
  URL 6
Cloud Developer community
  URL 6
common configuration
  data filters 56
  display business names 56
  overview 55
connections
  Salesforce 11
custom query 34

D

Data Integration community
  URL 6
Data Masking tasks
  relationship reconciliation strategies 48
data type reference
  overview 65
  Salesforce and transformation data types 65
data types
  source and target inconsistencies 44

E

external IDs
  for related Salesforce objects 55

F

flat tile targets
  truncated in outbound message tasks 57
full load
  and non-replicateable objects 44
  replication task load type 42

I

IDs
  types for objects 55
incremental load
  Replication task load type 43
  rules for running 44
indexes
  creating automatically when replicating Salesforce sources 40
Informatica Global Customer Support
  contact information 7
Informatica Intelligent Cloud Services
  web site 6

L

load types
  for replication 42
  full load 42
  incremental load 43

M

maintenance outages 7
Mapping tasks
  objects 30
  Salesforce lookups 34
  Salesforce sources 31
  Salesforce targets 32
mappings
  objects 25
  Salesforce lookups 30
  Salesforce sources 25
  Salesforce targets 28
masking
  advanced Salesforce options 54
  primary key chunking 54
Masking tasks
  overview 48
    relationship reconciliation strategies
      custom field lookup 49
      junction objects 51
      target owner name 51
  Salesforce limitations 52
  special handling of objects 53

N

non-replicateable objects
  loading 44
null updates to related objects 24

O

ODBC
  Synchronization task 15
outbound messages
  from Salesforce, starting tasks 57

P

precision
  source and target inconsistencies 44

R

related objects
  configuring external IDs 55
Replication task source
  advanced properties 40
replication tasks
  load types 42
Replication tasks
    configuration
      high precision calculations 46
      Salesforce API 46
      Salesforce Base64 encoded body size 46
  database target reset 44
  errors 44
  rules for running incremental loads 44
  Salesforce sources 40
requirements
  incremental load 44
rules and guidelines
  configuring Replication tasks 46
  Salesforce sources 17
  Salesforce targets 18

S

Salesforce
  connection properties 11
  data type changes, replication implications 44
  including deleted or archived source data 40
  starting tasks with outbound messages 57
Salesforce Connector
  connector example 9
  connector overview 8
Salesforce Connector *(continued)*
  tasks and object types 8
Salesforce IDs
  for objects 55
Salesforce source field metadata changes 45
scale
  source and target inconsistencies 44
sources
  configuring multiple-object sources 16, 27
status
  Informatica Intelligent Cloud Services 7
Synchronization tasks
  advanced options 20
  bulk API 22
  enable primary key chunking 24
  enable serial mode 23
  field mappings 19
  hard deletes 24
  including archived and deleted Salesforce data 17
  Salesforce sources 15
  Salesforce targets 17
  standard API 21
  upsert task operation 18
system status 7

T

trust site
  description 7

U

upgrade notifications 7

W

web site 6