



Informatica® PowerExchange for Kafka
10.5.1

User Guide for PowerCenter

Informatica PowerExchange for Kafka User Guide for PowerCenter

10.5.1

September 2021

© Copyright Informatica LLC 2018, 2021

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

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

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

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

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

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

Publication Date: 2021-09-24

Table of Contents

Preface	5
Informatica Resources.	5
Informatica Network.	5
Informatica Knowledge Base.	5
Informatica Documentation.	5
Informatica Product Availability Matrices.	6
Informatica Velocity.	6
Informatica Marketplace.	6
Informatica Global Customer Support.	6
Chapter 1: Introduction to PowerExchange for Kafka	7
PowerExchange for Kafka Overview.	7
Introduction to Kafka.	7
Chapter 2: PowerExchange for Kafka Configuration	8
PowerExchange for Kafka Configuration Overview.	8
Registering the PowerExchange for Kafka Plug-in.	8
Registering the PowerExchange for Kafka Plug-in from the Administrator Tool.	9
Registering the PowerExchange for Kafka Plug-in from the Command Line Interface.	9
Configuring JVM Options to Import JSON Schema.	10
Configuring JVM Options to Import Avro Schema.	10
Configuring SASL PLAIN Authentication for a Kafka Cluster.	11
Configuring the krb5.conf file to Read Data from or Write to a Kerberised Kafka Cluster (Optional). .	12
Configuring JVM Options to Access a Kerberised Kafka Cluster (Optional).	14
Chapter 3: Kafka Sources and Targets	15
Kafka Sources and Targets Overview.	15
Import Kafka Source and Target Definitions.	15
Sample Schema Files.	21
Chapter 4: Kafka Mappings	22
Kafka Mappings Overview.	22
Kafka Mapping Example.	22
Chapter 5: Kafka Sessions	25
Kafka Sessions Overview.	25
Kafka Connections.	26
PowerExchange for Kafka Connections.	26
Configuring a Kafka Connection.	27
Configure Kafka Source Session Properties.	28

Configure Kafka Target Session Properties.	29
Configure Message Recovery Strategy for a Kafka Session.	29
Rules and Guidelines for Kafka Sessions.	30
Appendix A: Kafka Datatype Reference.	32
Data Type Reference Overview.	32
Kafka and Transformation Data Types.	32
JSON Kafka Data Types and Transformation Data Types.	33
Avro Kafka Data Types and Transformation Data Types.	34
Index.	35

Preface

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

Informatica Resources

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

Informatica Network

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

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

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

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

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

Informatica Documentation

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

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

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

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

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

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

Informatica Global Customer Support

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

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

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

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

CHAPTER 1

Introduction to PowerExchange for Kafka

This chapter includes the following topics:

- [PowerExchange for Kafka Overview, 7](#)
- [Introduction to Kafka, 7](#)

PowerExchange for Kafka Overview

You can use PowerExchange for Kafka for connectivity between PowerCenter and Kafka.

PowerExchange for Kafka sources and targets represents data in a Kafka stream.

You can use Kafka topics as sources and targets in mappings. When you use Kafka topics in mappings, you must configure properties specific to Kafka.

Example

You run the IT department of a major bank that has millions of customers. You want to monitor network activity in real time. You need to collect network activity data from various sources such as firewalls or network devices to improve security and prevent attacks. The network activity data includes Denial of Service (DoS) attacks and failed login attempts made by customers. The network activity data is written to Kafka queues.

You can use PowerExchange for Kafka to read network activity data from Kafka topics in JSON format and write the data to a target for processing the network activity data..

Introduction to Kafka

Kafka is an open-source scalable, fault tolerant, and distributed data streaming platform. You can process large amounts of data quickly by using Kafka. It can serve as an interim staging area for data that will be consumed by various types of downstream consumer applications.

Kafka runs as a cluster comprised of one or more servers each of which is called a broker. Kafka brokers stream data in the form of messages. These messages are published to a topic.

Kafka topics are divided into partitions. This gives better throughput and could be used to scale the number of messages processed. Message ordering is guaranteed only within partitions.

CHAPTER 2

PowerExchange for Kafka Configuration

This chapter includes the following topics:

- [PowerExchange for Kafka Configuration Overview, 8](#)
- [Registering the PowerExchange for Kafka Plug-in, 8](#)
- [Configuring JVM Options to Import JSON Schema, 10](#)
- [Configuring JVM Options to Import Avro Schema, 10](#)
- [Configuring SASL PLAIN Authentication for a Kafka Cluster, 11](#)
- [Configuring the krb5.conf file to Read Data from or Write to a Kerberised Kafka Cluster \(Optional\), 12](#)
- [Configuring JVM Options to Access a Kerberised Kafka Cluster \(Optional\), 14](#)

PowerExchange for Kafka Configuration Overview

PowerExchange for Kafka installs with Informatica services.

If you upgrade from a previous version, you must register the PowerExchange for Kafka plug-in with the PowerCenter repository.

To import the schema of Kafka topic in JSON or Avro format, apply the Informatica EBF-16426 at <https://tsftp.informatica.com/> to PowerCenter Client from the following location:

```
/updates/Informatica10/10.4.0
```

For more information, contact Informatica Global Customer Support.

Note: Ensure that you install the Apache Kafka cluster and the Informatica Services in different nodes.

Registering the PowerExchange for Kafka Plug-in

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

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

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

The name of the plug-in file for PowerExchange for Kafka is `kafka_Plugin.xml`.

Registering the PowerExchange for Kafka Plug-in from the Administrator Tool

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

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

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

8. Run the PowerCenter Repository Service in normal mode.

Registering the PowerExchange for Kafka Plug-in from the Command Line Interface

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

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

The `RegisterPlugin` command uses the following syntax:

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

3. Find `kafka_Plugin.xml` in the following directory:

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

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

The `RegisterPlugin` command uses the following syntax:

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

Configuring JVM Options to Import JSON Schema

If you configure a session to read data from or write data to a Kafka topic in JSON format, you must configure the JVM options for the PowerCenter Client and the PowerCenter Integration Service to specify the path of the Hadoop distribution package.

Configuring JVM Options for the PowerCenter Client

You can configure the `ccijvmoptions.ini` file for the PowerCenter Client.

1. Ensure that you have applied the Informatica EBF-16426 to PowerCenter Client.
2. Access the `ccijvmoptions.ini` file from the following location:
<PowerCenter Installation Directory>\clients\PowerCenterClient\client\bin
3. Add the following property to the `ccijvmoptions.ini` file:

```
-DINFA_HADOOP_DIST_DIR=<Informatica installation directory>\clients  
\PowerCenterClient\hadoop\CDH_6.3
```

4. Restart the PowerCenter Client.

Configuring JVM Options to Import Avro Schema

If you configure a session to read data from or write data to a Kafka topic in Avro format, you must configure the JVM options for the PowerCenter Client and the PowerCenter Integration Service to enable Avro data format and specify the path of the Hadoop distribution package.

Configuring JVM Options for the PowerCenter Client

You can configure the `ccijvmoptions.ini` file for the PowerCenter Client.

1. Ensure that you have applied the Informatica EBF-16426 to PowerCenter Client.
2. Access the `ccijvmoptions.ini` file from the following location:
<PowerCenter Installation Directory>\clients\PowerCenterClient\client\bin
3. Add the following properties to the `ccijvmoptions.ini` file:

```
-DENABLE_AVRO_FORMAT_FOR_KAFKA=true  
-DINFA_HADOOP_DIST_DIR=<Informatica installation directory>\clients\PowerCenterClient  
\hadoop\CDH_6.3
```

4. Restart the PowerCenter Client.

Configuring JVM Options for the PowerCenter Integration Service

You can configure the JVM options for the PowerCenter Integration Service to enable Avro data format and specify the path of the Hadoop distribution package from the Administrator Console.

1. In the Administrator tool Domain Navigator, select the PowerCenter Integration Service for which you want to set the kerberised Kafka settings.
2. Click the **Processes** tab on the right pane.
3. Click **Edit** in the **Custom Properties** section.

- Define the following properties and values:

Property	Value
JVMOption2	-DENABLE_AVRO_FORMAT_FOR_KAFKA=true
JVMOption3	-DINFA_HADOOP_DIST_DIR=<Informatica installation directory>\services \shared\hadoop\CDH_6.3

- Restart the PowerCenter Integration Service.

Configuring SASL PLAIN Authentication for a Kafka Cluster

To read from or write to a Kafka cluster with SASL PLAIN authentication, you can either configure the Kafka connection properties or in the source or target session properties. If you have already configured SASL PLAIN authentication in the Kafka connection properties, you can choose to override those properties from the source or target session properties.

Configure SASL PLAIN authentication in the following properties:

Kafka connection

Configure the **Additional Connection Properties** property in a Kafka connection and specify the value in the following format:

```
security.protocol=SASL_SSL,sasl.mechanism=PLAIN,sasl.jaas.config=org.apache.kafka.com
mon.security.plain.PlainLoginModule required username=<username> password=<password>
```

In the Security Configuration Section, select **One-Way** as the **SSL Mode** and specify the **SSL TrustStore File Path** and **SSL TrustStore Password**.

Sources

Configure the **Consumer Configuration Properties** property in the source session properties to override the value specified in the **Additional Connection Properties** property in a Kafka connection. Specify the value in the following format:

```
security.protocol=SASL_SSL,sasl.mechanism=PLAIN,sasl.jaas.config=org.apache.kafka.com
mon.security.plain.PlainLoginModule required username=<username> password=<password>
```

Targets

Configure the **Producer Configuration Properties** property in the target session properties to override the value specified in the **Additional Connection Properties** property in a Kafka connection. Specify the value in the following format:

```
security.protocol=SASL_SSL,sasl.mechanism=PLAIN,sasl.jaas.config=org.apache.kafka.com
mon.security.plain.PlainLoginModule required username=<username> password=<password>
```

Configuring the krb5.conf file to Read Data from or Write to a Kerberised Kafka Cluster (Optional)

To read from or write to a Kerberised Kafka cluster, configure the default realm, KDC, and Kafka source or target session properties.

Before you read from or write to a Kerberised Kafka cluster, perform the following tasks:

1. Ensure that you have the `krb5.conf` file for the Kerberised Kafka cluster.
2. Configure the default realm and KDC. If the default `/etc/krb5.conf` file is not configured or you want to change the configuration, add the following lines to the `/etc/krb5.conf` file:

```
[libdefaults]
default_realm = <REALM NAME>
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true

[realms]
<REALM NAME> = {
kdc = <Location where KDC is installed>
admin_server = <Location where KDC is installed>
}

[domain_realm]
.<domain name or hostname> = <KERBEROS DOMAIN NAME>
<domain name or hostname> = <KERBEROS DOMAIN NAME>
```

3. To pass a static JAAS configuration file into the JVM using the `java.security.auth.login.config` property at runtime, perform the following tasks:
 - a. Ensure that you have JAAS configuration file.
For information about creating JAAS configuration and configuring Keytab for Kafka clients, see the Apache Kafka documentation at <https://kafka.apache.org/0101/documentation/#security>

For example, the JAAS configuration file can contain the following lines of configuration:

```
//Kafka Client Authentication. Used for client to kafka broker connection
KafkaClient {
com.sun.security.auth.module.Krb5LoginModule required
doNotPrompt=true
useKeyTab=true
storeKey=true
keyTab="<path to Kafka keytab file>/<Kafka keytab file name>"
principal="<principal name>"
client=true
};
```

- b. Place the JAAS config file and keytab file in the same location on all the nodes.
Informatica recommends that you place the files in a location that is accessible to all the nodes in the cluster. Example: `/etc` or `/temp`
- c. Configure the following properties:

Kafka connection

Configure the **Kerberos Configuration Properties** property in a Kafka connection and specify the value in the following format:

```
security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI
```

Sources

Configure the **Consumer Configuration Properties** property in the source session properties to override the value specified in the **Kerberos Configuration Properties** property in a Kafka connection. Specify the value in the following format:

```
security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI
```

Targets

Configure the **Producer Configuration Properties** property in the target session properties to override the value specified in the **Kerberos Configuration Properties** property in a Kafka connection. Specify the value in the following format:

```
security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI
```

4. To embed the JAAS configuration in the `sasl.jaas.config` configuration property, configure the following properties:

Kafka connection

Configure the **Kerberos Configuration Properties** property in a Kafka connection and specify the value in the following format:

```
security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI,  
sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required  
useKeyTab=true  
storeKey=true doNotPrompt=true serviceName="<service_name>" keyTab="<location of  
keytab file>"  
client=true principal="<principal_name>;
```

Sources

Configure the **Consumer Configuration Properties** property in the source session properties to override the value specified in the **Kerberos Configuration Properties** property in a Kafka connection. Specify the value in the following format:

```
security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI,  
sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required  
useKeyTab=true  
storeKey=true doNotPrompt=true serviceName="<service_name>" keyTab="<location of  
keytab file>"  
client=true principal="<principal_name>;
```

Targets

Configure the **Producer Configuration Properties** property in the target session properties to override the value specified in the **Kerberos Configuration Properties** property in a Kafka connection. Specify the value in the following format:

```
security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI,  
sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required  
useKeyTab=true  
storeKey=true doNotPrompt=true serviceName="<service_name>" keyTab="<location of  
keytab file>"  
client=true principal="<principal_name>;
```

Configuring JVM Options to Access a Kerberised Kafka Cluster (Optional)

If you configure a session to read data from or write data to a Kafka topic in a Kerberised Kafka cluster, you must configure the JVM Options for the PowerCenter Client and the PowerCenter Integration Service to specify the path of the `krb5.conf` and `kafka_client_jaas.conf` file.

Configuring JVM Options for the PowerCenter Client

You can configure the `ccijvmoptions.ini` file for the PowerCenter Client.

1. Ensure that you have the `krb5.conf` and `kafka_client_jaas.conf` file for the Kerberised Kafka cluster.
2. Access the `ccijvmoptions.ini` file from the following location:
<PowerCenter Installation Directory>\clients\PowerCenterClient\client\bin
3. Add the following properties to the `ccijvmoptions.ini` file:

```
-Djava.security.auth.login.config=complete path and name of the  
kafka_client_jaas.conf  
-Djava.security.krb5.conf=complete path and name of the krb5.conf
```

4. Restart the PowerCenter Client.

Configuring JVM Options for the PowerCenter Integration Service

You can configure the JVM Options for the PowerCenter Integration Service to specify the path of the `krb5.conf` and `kafka_client_jaas.conf` file from the Administrator Console.

1. Ensure that you have the `krb5.conf` and `kafka_client_jaas.conf` file for the Kerberised Kafka cluster.
2. In the Administrator tool Domain Navigator, select the PowerCenter Integration Service for which you want to set the kerberised Kafka settings.
3. Click the **Processes** tab on the right pane.
4. Click **Edit** in the **Custom Properties** section.
5. Define the following properties and values:

Property	Value
JVMOption4	-Djava.security.auth.login.config=<complete path and name of the kafka_client_jaas.conf file>
JVMOption5	-Djava.security.krb5.conf=<complete path and name of the krb5.conf file>

For example,

```
JVMOption4=-Djava.security.auth.login.config=/etc/kafka/kafka_client_jaas.conf  
JVMOption5=-Djava.security.krb5.conf=/etc/kafka/krb5.conf
```

6. Restart the PowerCenter Integration Service.

CHAPTER 3

Kafka Sources and Targets

This chapter includes the following topics:

- [Kafka Sources and Targets Overview, 15](#)
- [Import Kafka Source and Target Definitions, 15](#)
- [Sample Schema Files, 21](#)

Kafka Sources and Targets Overview

You can create a mapping with a Kafka source to read data from a Kafka topic. You can create a mapping with any source and a Kafka target to write data to a Kafka topic.

A Kafka source or target represents data in a Kafka stream. Create a PowerExchange for Kafka source or target to read from or write to Apache Kafka brokers.

Kafka runs as a cluster comprised of one or more servers each of which is called a broker. Kafka brokers stream data in the form of messages. These messages are published to a topic.

When you import a Kafka source or target, select the topic name that you want to read from or write to. You can specify the topic name or use a regular expression for the topic name pattern only when you read from Kafka. To subscribe to multiple topics that match a pattern, you can specify a regular expression. When you run the application on the cluster, the pattern matching is done against topics before the application runs. If you add a topic with a similar pattern when the application is already running, the application will not read from the topic.

When you import a Kafka topic, you can specify the format in which the PowerExchange for Kafka reads data from or writes data to a Kafka topic. You can specify JSON or Avro as format. When you specify JSON format, you must provide a sample JSON schema file. When you specify Avro format, provide a sample Avro schema as a `.avsc` file.

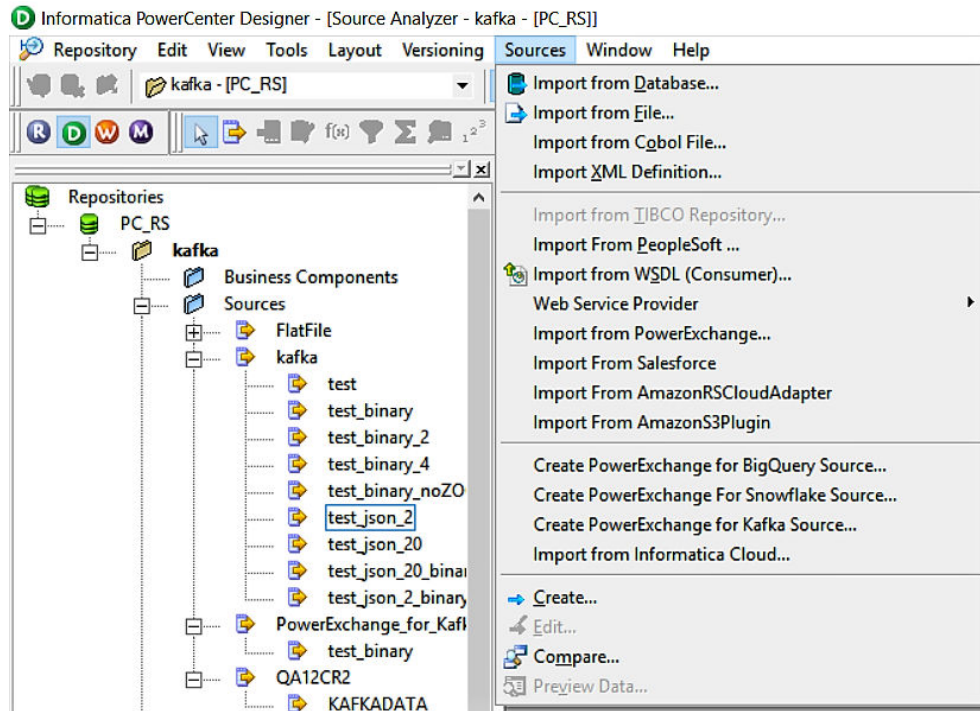
Import Kafka Source and Target Definitions

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

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

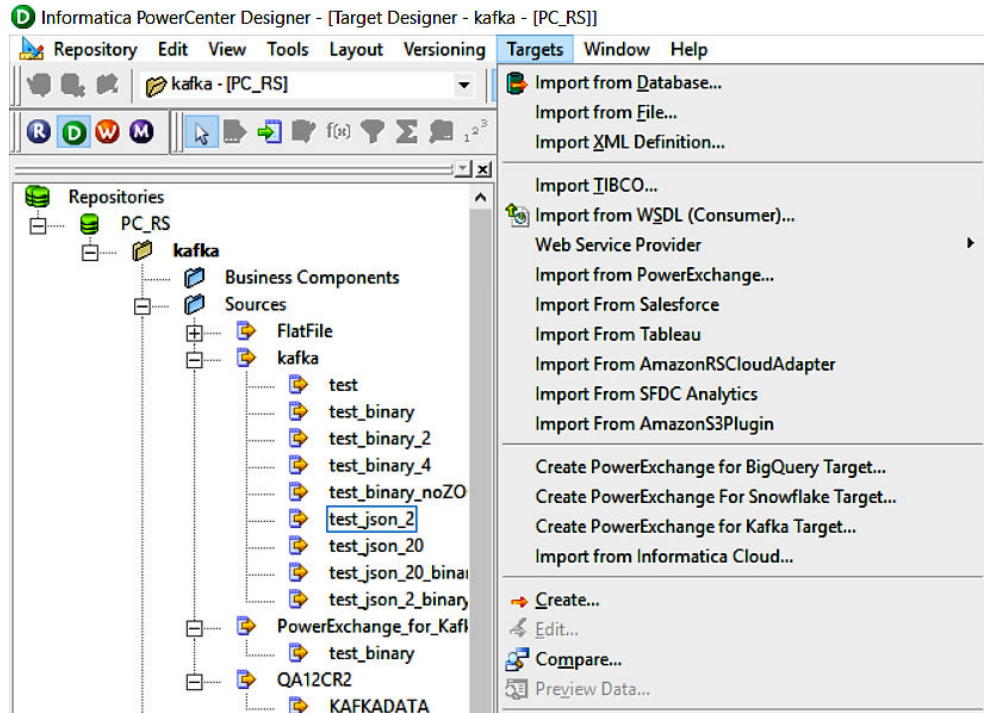
1. Start PowerCenter Designer, and connect to a PowerCenter repository configured with a Kafka instance.

2. Open a source or target folder.
3. Select **Source Analyzer** or **Target Designer**.
 - In the Source Analyzer, click **Sources > Create PowerExchange for Kafka Source** as shown in the following image:



The **Kafka Connection** wizard appears.

- In the Target Analyzer, click **Targets > Create PowerExchange for Kafka Target** as shown in the following image:



The **Kafka Connection** dialog box appears.

4. Configure the following connection parameters:

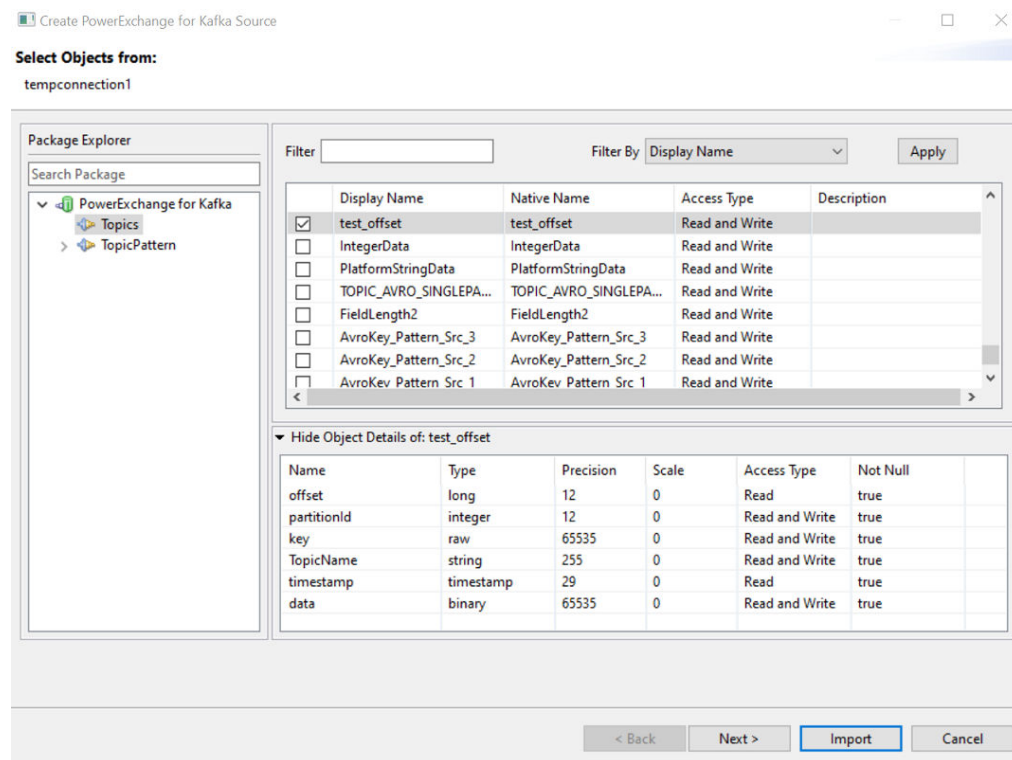
Connection Parameter	Description
Kafka Broker List	The IP address and port combinations of the Kafka messaging system broker list. The IP address and port combination has the following format: <IP Address>:<port> You can enter multiple comma-separated IP address and port combinations.
Retry Timeout in seconds	Number of seconds the Integration Service attempts to reconnect to the Kafka broker to write data. If the source or target is not available for the time you specify, the mapping execution stops to avoid any data loss. Default is 180 seconds.
Kafka Broker Version	Select Apache 0.10.1.1 & above as the Kafka messaging broker version.

Connection Parameter	Description
Additional Connection Properties	<p>Comma-separated list of connection properties to connect to the Kafka broker.</p> <p>To connect to a Kafka broker over SSL, ensure that you set the following properties:</p> <ul style="list-style-type: none"> - security.protocol - ssl.truststore.location - ssl.truststore.password <p>If the <code>ssl.client.auth</code> parameter is set to <code>required</code> or <code>requested</code>, ensure that you set the following properties:</p> <ul style="list-style-type: none"> - ssl.keystore.location - ssl.keystore.password - ssl.key.password <p>For example, <code>security.protocol=SSL,ssl.truststore.location=<complete path and name of the client.truststore.jks file>,ssl.truststore.password=<client truststore password>,ssl.keystore.location=<complete path and name of the client.keystore.jks file>,ssl.keystore.password=<client keystore password>,ssl.key.password=<password for the individual key></code></p> <p>You can also specify the configuration properties required to connect to a Kerberised Kafka cluster.</p> <p>Specify the following property:</p> <pre>security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI</pre>
SSL Mode	<p>Specifies whether the PowerCenter Integration Service establishes a secure connection to the Kafka broker. You can select one of the following options:</p> <ul style="list-style-type: none"> - disabled. The PowerCenter Integration Service establishes an unencrypted connection to the Kafka broker. - One-Way. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker using truststore file and truststore password. - Two-Way. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker using truststore file and truststore password.
SSL TrustStore File Path	<p>Applicable only if you select One-Way or Two-Way as the SSL mode.</p> <p>The complete path and file name of the truststore file. The truststore file contains the SSL certificate that the Kafka cluster validates against the Kafka broker certificate.</p>
SSL TrustStore Password	<p>Applicable only if you select One-Way or Two-Way as the SSL mode.</p> <p>The password for the truststore file.</p>
SSL KeyStore File Path	<p>Applicable only if you select Two-Way as the SSL mode.</p> <p>The complete path and name of the Java keystore file. The keystore file contains the certificate that the Kafka broker validates against the Kafka cluster certificate</p>
SSL KeyStore Password	<p>Applicable only if you select Two-Way as the SSL mode.</p> <p>The password for the keystore file.</p>
Additional Security Properties	<p>Not applicable for PowerExchange for Kafka.</p>

5. Click **Test** to test the connection.
 6. Click **Finish** to add the connection.
- The **Select Objects from** tab appears.

7. Select the **Topics** package in **Package Explorer**.

A list of topic appears as shown in the following image:

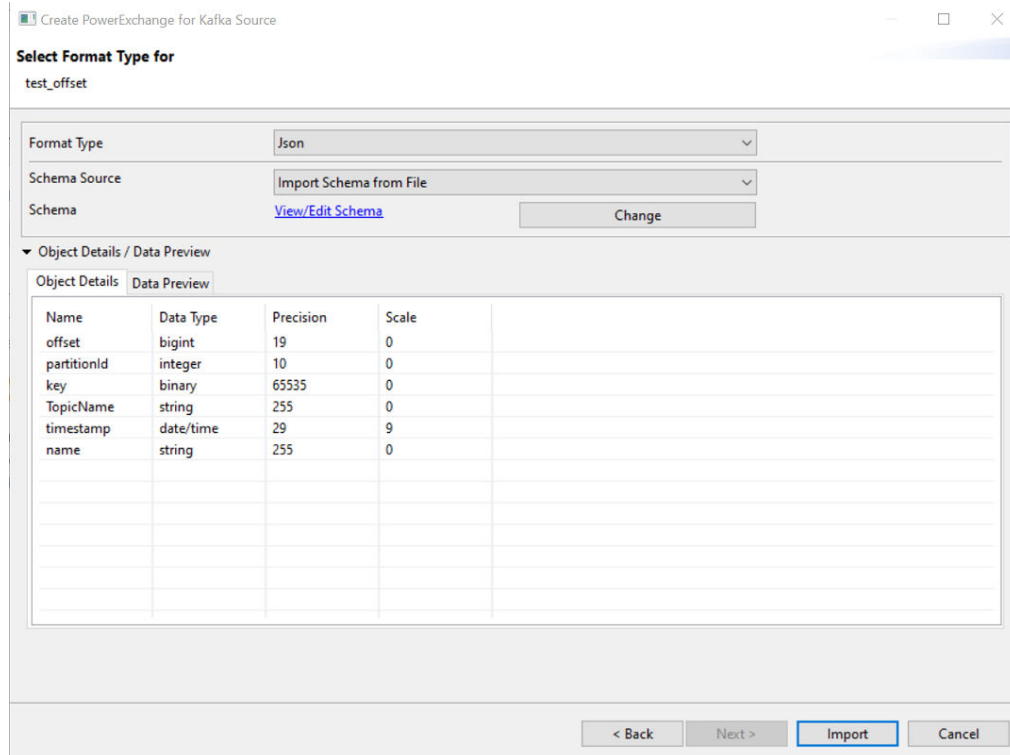


8. Select the topic that you want to import, and then click **Next**.

Note: You cannot select topics from the **Topics Pattern** package.

The **Select Format Type for** dialog box appears.

The following image shows the **Select Format Type for** dialog box:



9. In the **Format Type** list, select **Json** or **Avro** to import data from Kafka topics in JSON format or Avro format.
Note: You cannot import a PowerExchange for Kafka source or target definition in Flat or XML format.
10. If you select **Json** or **Avro** as the **Format Type**, you must configure the following format properties:

Property	Description
Schema Source	Specifies the mode to import schema for the Kafka topic. Select Import Schema from File to import schema from a schema definition file in your local machine. You cannot import a schema for a Kafka topic using the Read Data from File option.
Schema	When you select the Import Schema from File option in the Schema Source drop-down list, you must choose a schema definition file on your local machine.

11. Optionally, click **Data Preview** to preview data.
To view the topic metadata, select the topic, and double-click the topic name.
Note: You cannot select multiple topics in a Kafka source or target definition.
12. Click **Import**.
Note: If you click **Import** in the **Select Objects from** dialog box, PowerExchange for Kafka imports the data from the Kafka topic in binary format.

Sample Schema Files

The Kafka sources and targets in a session read or write data in JSON or Avro format. The following examples contain samples for each schema format.

Sample JSON Schema

When you import a Kafka source or target definition, specify the format in which the PowerExchange for Kafka reads or writes data. When you specify JSON format, provide a sample JSON file.

The following schema is a sample JSON schema:

```
{"id" : 1, "name" : "sample"}
```

Note: The JSON schema must be defined in a single line. If you define a JSON schema that contains more than one line, the session fails.

Sample Avro Schema

When you import a Kafka source or target definition, specify the format in which the PowerExchange for Kafka reads or writes data. When you specify Avro format, provide a sample Avro file.

The following schema is a sample Avro schema:

```
{
  "type": "record",
  "name": "Tree",
  "fields": [{
    "name": "treestring",
    "type": "string"
  },
  {
    "name": "children",
    "type": ["null", "string"],
    "default": null
  }
]}
```

CHAPTER 4

Kafka Mappings

This chapter includes the following topics:

- [Kafka Mappings Overview, 22](#)
- [Kafka Mapping Example, 22](#)

Kafka Mappings Overview

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

Kafka Mapping Example

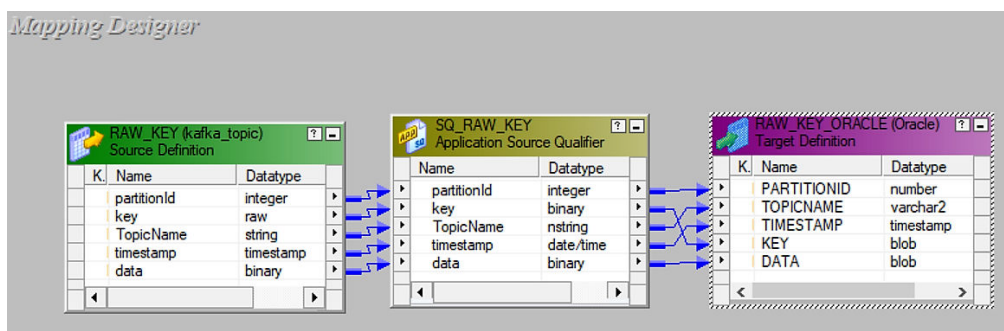
You run the IT department of a major bank that has millions of customers. You want to monitor network activity in real time. You need to collect network activity data from various sources such as firewalls or network devices to improve security and prevent attacks. The network activity data includes Denial of Service (DoS) attacks and failed login attempts made by customers. The network activity data is written to Kafka queues.

Create a mapping to read the network activity data from Kafka topics and write the data to Oracle for processing the network activity data.

The following procedure shows how to move data from the Kafka topic to Oracle:

1. Import a Kafka source.
2. Import an Oracle target.

3. Create a mapping with the Kafka source and an Oracle target.
The following image shows the example mapping:



The mapping contains the following objects:

Source Definition

The mapping source definition is a Kafka topic that contains the network activity data.

In the Source Analyzer, import the Kafka topic that you want to read. The PowerCenter Integration Service reads network activity data from the Kafka topic.

The following table describes the structure of the Kafka source definition:

Field Name	Data Type
partitionId	Integer
key	Raw
TopicName	String
timestamp	Timestamp
data	Binary

Target Definition

The mapping contains an Oracle target definition. In the Target Designer, import an Oracle target definition.

The following table describes the structure of the Oracle target definition:

The following table describes the structure of the Oracle target definition:

Field Name	Data Type
PARTITIONID	Number(p,s)
TOPICNAME	varchar2
TIMESTAMP	Varchar2

Field Name	Data Type
DATA	Blob
KEY	Blob

Link ports between the Kafka Source Qualifier and Oracle target definition to create a flow of data. In the **Workflow Manager**, create a session and add the mapping to the session. Create a workflow and add the session to the workflow.

When you run the workflow, the data is read from the Kafka queue and written to the Oracle target. You can then run queries on the Oracle database to analyze the network activity data.

CHAPTER 5

Kafka Sessions

This chapter includes the following topics:

- [Kafka Sessions Overview, 25](#)
- [Kafka Connections, 26](#)
- [Configure Kafka Source Session Properties, 28](#)
- [Configure Kafka Target Session Properties, 29](#)
- [Configure Message Recovery Strategy for a Kafka Session, 29](#)
- [Rules and Guidelines for Kafka Sessions, 30](#)

Kafka Sessions Overview

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

You must configure a Kafka connection in the **Workflow Manager** to extract data from or load data to Kafka topics. You can define properties in a session to determine how the PowerCenter Integration Service must extract data from or load data to a Kafka topic.

Kafka Connections

The Kafka connection is a Messaging connection. Use the Kafka connection to access an Apache Kafka broker as a source or a target. You can create and manage a Kafka connection in the PowerCenter Workflow Manager or infacmd. Create a Kafka connection to read data from and write data to Apache Kafka brokers.

PowerExchange for Kafka Connections

When you configure a Kafka connection, you define the connection attributes that the PowerCenter Integration Service uses to connect to the Kafka cluster. You can configure SSL properties to connect to a Kafka Broker over SSL.

The following table describes the Kafka connection properties:

Property	Description
Kafka Broker List	The IP address and port combinations of the Kafka messaging system broker list. The IP address and port combination has the following format: <IP Address>:<port> You can enter multiple comma-separated IP address and port combinations.
Retry Timeout in seconds	Number of seconds the Integration Service attempts to reconnect to the Kafka broker to write data. If the source or target is not available for the time you specify, the mapping execution stops to avoid any data loss. Default is 180 seconds.
Kafka Broker Version	Select Apache 0.10.1.1 and above as the Kafka messaging broker version.
Additional Connection Properties	Comma-separated list of connection properties to connect to the Kafka broker. To connect to a Kafka broker over SSL, ensure that you set the following properties: <ul style="list-style-type: none">- security.protocol- ssl.truststore.location- ssl.truststore.password If the <code>ssl.client.auth</code> parameter is set to <code>required</code> or <code>requested</code> , ensure that you set the following properties: <ul style="list-style-type: none">- ssl.keystore.location- ssl.keystore.password- ssl.key.password For example, <code>security.protocol=SSL,ssl.truststore.location=<complete path and name of the client.truststore.jks file>,ssl.truststore.password=<client truststore password>,ssl.keystore.location=<complete path and name of the client.keystore.jks file>,ssl.keystore.password=<client keystore password>,ssl.key.password=<password for the individual key></code> You can also specify the configuration properties required to connect to a Kerberised Kafka cluster. Specify the following property: <code>security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI</code>

Property	Description
SSL Mode	Specifies whether the PowerCenter Integration Service establishes a secure connection to the Kafka broker. You can select one of the following options: <ul style="list-style-type: none"> - disabled. The PowerCenter Integration Service establishes an unencrypted connection to the Kafka broker. - require. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker without verifying the identity of the server. - one-way. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker using truststore file and truststore password. - two-way. The PowerCenter Integration Service establishes an encrypted connection to the Kafka broker using truststore file and truststore password.
SSL TrustStore File Path	Applicable only if you select one-way or two-way as the SSL mode. The complete path and file name of the truststore file. The truststore file contains the SSL certificate that the Kafka cluster validates against the Kafka broker certificate.
SSL TrustStore Password	Applicable only if you select one-way or two-way as the SSL mode. The password for the truststore file.
SSL KeyStore File Path	Applicable only if you select two-way as the SSL mode. The complete path and name of the Java keystore file. The keystore file contains the certificate that the Kafka broker validates against the Kafka cluster certificate
SSL KeyStore Password	Applicable only if you select two-way as the SSL mode. The password for the keystore file.
Additional Security Properties	Optional. Comma-separated list of connection properties to connect to the Kafka broker in a secured way. For example, <code>security.protocol=SASL_PLAINTEXT, sasl.kerberos.service.name=<kerberos name>, sasl.mechanism=GSSAPI, sasl.jaas.config=com.sun.security.auth.module.Krb5LoginModule required useKeyTab=true doNotPrompt=true storeKey=true client=true keyTab="<Keytab Location>" principal="<principal>"</code> ;

Configuring a Kafka Connection

Configure a Kafka connection in the **Workflow Manager** to define the connection attributes that the PowerCenter Integration Service uses to connect to a Kafka cluster.

1. In the **Workflow Manager**, click **Connections > Application**.
The **Application Connection Browser** dialog box appears.
2. Click **New**.
The **Select Subtype** dialog box appears.
3. Select **PowerExchange for Kafka** and click **OK**.
The **Application Connection Editor** dialog box appears.
4. Enter a name for the Kafka connection.
5. Enter the Kafka connection attributes.
6. Click **OK** to create a Kafka connection.

Configure Kafka Source Session Properties

You can configure the session properties for a Kafka source on the **Mapping** tab. Define the properties for the source instance in the session.

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

Property	Description
Start position offset	<p>The time from which the Kafka source starts reading Kafka messages from a Kafka topic.</p> <p>You can select one of the following options:</p> <ul style="list-style-type: none"> - Custom. Read messages from a specific time. - Earliest. Read the earliest messages available on the Kafka topic. - Latest. Read messages received by the Kafka topic after the mapping has been deployed.
Connector Mode	<p>Specifies the mode to read data from the Kafka source.</p> <p>You can select one of the following modes:</p> <ul style="list-style-type: none"> - Batch. The PowerCenter Integration Service reads the messages available in a Kafka topic based on the specified position offset. After the PowerCenter Integration Service reads the data, the session terminates. When you configure a session for recovery, the PowerCenter Integration Service reads data from the last checkpoint and stops reading data till the offset when the session recovery started. - Streaming. The PowerCenter Integration Service reads the messages available in a Kafka topic in real time. The PowerCenter Integration Service does not terminate the session and continues to read messages from the Kafka topic. You must manually terminate the workflow.
Custom Start Position Timestamp	<p>The time in GMT from which the Kafka source starts reading Kafka messages from a Kafka topic.</p> <p>Specify a time in the following format:</p> <pre>yyyy-mm-dd HH24:mm:ss[.fffffffff]</pre> <p>The milliseconds are optional.</p> <p>This property is applicable only when you select Custom as the Start position offset.</p>
Consumer Configuration Properties	<p>The configuration properties for the Kafka consumer.</p> <p>If you read data from a Kafka cluster that is configured for Kerberos authentication, specify the following property to override the Additional Connection Properties specified in the Kafka connection:</p> <pre>security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI</pre> <p>For more information about the Kafka consumer configuration properties, see the Kafka documentation.</p>
Recover from last state	<p>If you select this option, the PowerCenter Integration Service recovers a stopped, aborted, or terminated session from the last checkpoint.</p> <p>Default is selected.</p>
Topic Pattern	<p>Specify a regular expression pattern for the topic name that you want to read from. Use the regular expression syntax guidelines to specify the pattern.</p> <p>For more information about how to use topic patterns to read data from a Kafka source, see the Knowledge Base article 565201.</p>
Tracing Level	<p>Amount of detail displayed in the session log for the transformation. You can choose Normal, Verbose Initialization, or Verbose Data.</p> <p>Default is Normal.</p>

Configure Kafka Target Session Properties

You can configure the session properties for a Kafka target on the **Mapping** tab. Define the properties for the target instance in the session.

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

Property	Description
Metadata Fetch Timeout in milliseconds	Not applicable for PowerExchange for Kafka.
Batch Flush Time in milliseconds	The interval after which the data is published to the target. Not applicable for PowerExchange for Kafka.
Batch Flush Size in bytes	The batch size of the events after which the PowerCenter Integration Service writes data to the target. Not applicable for PowerExchange for Kafka.
Producer Configuration Properties	The configuration properties for the producer. If you write data to a Kafka cluster that is configured for Kerberos authentication, include the following property to override the Additional Connection Properties specified in the Kafka connection: <code>security.protocol=SASL_PLAINTEXT,sasl.kerberos.service.name=kafka,sasl.mechanism=GSSAPI</code> For more information about the Kafka producer configuration properties, please see the Kafka documentation.

Configure Message Recovery Strategy for a Kafka Session

When you configure message recovery for a real-time session, the PowerCenter Integration Service can recover unprocessed messages from a failed session. When you enable message recovery for a real-time session, the PowerCenter Integration Service stores source messages or message IDs in a recovery file or recovery table. If the session fails, run the session in recovery mode to recover the messages that the PowerCenter Integration Service did not process.

When you configure a session for recovery, you can recover the session when you recover a workflow, or you can recover the session without running the rest of the workflow.

When you configure a session, you can configure the **Recovery Strategy** property on the **Properties** page of the Session task.

You can select one of the following recovery strategies:

- **Resume from the last checkpoint.** The PowerCenter Integration Service saves the session state of operation and maintains target recovery files or recovery tables. If the session aborts, stops, or terminates, the PowerCenter Integration Service uses the saved recovery information to resume the session from the point of interruption. You cannot configure a session with a resume strategy if it uses mapping variables.

- **Restart task.** The PowerCenter Integration Service runs the session again when it recovers the workflow. When you recover with restart task, you might need to remove the partially loaded data in the target or design a mapping to skip the duplicate rows.
- **Fail task and continue workflow.** When the PowerCenter Integration Service recovers a workflow, it does not recover the session. The session status becomes failed, and the PowerCenter Integration Service continues running the workflow.

Steps to Enable Message Recovery

Complete the following steps to enable message recovery for sessions:

1. In the session properties, select Resume from Last Checkpoint as the recovery strategy.
2. Specify a recovery storage directory for the PowerCenter Integration Service.

The PowerCenter Integration Service stores messages in the location indicated by the recovery storage directory. The default value recovery storage directory is `$PMStorageDir`.

The PowerCenter Integration Service uses the following format for the recovery file name:

```
pmgmd_cciFile_<AppIDFolderID>-<GroupIDOwnerTypeID>-<PartitionIDPluginID>-<RepGIDSessionID>-<WidgetIDWorkflowID>
```

For example, the PowerCenter Integration Service creates the recovery file with the following file name:
`pmgmd_cciFile_240100d96a0936-911c-4bba-bf7e-05a3dbedb7201224217920`

Rules and Guidelines for Kafka Sessions

Use the following rules and guidelines when you configure a Kafka session:

- You cannot configure partitioning to read data from a Kafka source.
- When you select JSON as the format type, you cannot read hierarchical data from a Kafka source or write hierarchical data to a Kafka target.
- When you use a Kafka source or target definition in a mapping, you cannot configure multiple pipelines.
- When you read data from a Kafka topic and write data to an Oracle target table, you must not select **Bulk** as the **Target Load Type** under the target session properties. Otherwise, the session fails.
- When you read data from a Kafka topic or use a topic pattern and the offset of the last checkpoint is deleted during message recovery, you must provide the following property in the **Consumer Configuration Properties** to recover the messages from the next available offset:
`auto.offset.reset=earliest`. Otherwise, the mapping reads data from the latest offset available.
- When you remove the link for a port between the Kafka source definition and the Kafka Source Qualifier transformation, you must remove the port from the Kafka Source Qualifier transformation. Otherwise, the mapping fails with the following error:
`com.informatica.powercenter.sdk.SDKException`
- When you read data from a Kafka topic and write data to any target, you must select **Resume from Last Checkpoint** as the **Recovery Strategy**.
- When you write data to a Kafka target and enable message recovery for the session, the PowerCenter Integration Service only supports file-based recovery.
- When you write data to a Kafka target and enable message recovery for the session, the PowerCenter Integration Service loads data from the last checkpoint stored in the recovery file.

- When you read data from or write data to a Kafka topic, ensure that you do not map the **offset** port in the mapping.

APPENDIX A

Kafka Datatype Reference

This appendix includes the following topics:

- [Data Type Reference Overview, 32](#)
- [Kafka and Transformation Data Types, 32](#)
- [JSON Kafka Data Types and Transformation Data Types, 33](#)
- [Avro Kafka Data Types and Transformation Data Types, 34](#)

Data Type Reference Overview

PowerCenter uses the following data types in Kafka mappings:

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

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

Kafka and Transformation Data Types

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

The following table lists the Kafka data types that PowerCenter supports and the corresponding transformation data types:

Kafka Data Type	Transformation Data Type	Range and Description
BINARY	Binary	1 to 104,857,600 bytes
INTEGER	Integer	Precision 10, scale 0

Kafka Data Type	Transformation Data Type	Range and Description
RAW	Binary	1 to 104,857,600 bytes
STRING	Nstring	1 to 104,857,600 characters
TIMESTAMP	Date/Time	Date and time values. (precision to the nanosecond)

JSON Kafka Data Types and Transformation Data Types

JSON Kafka data types map to transformation data types that the PowerCenter Integration Service uses to move data across platforms.

The following table lists the JSON Kafka data types that the PowerCenter Integration Service supports and the corresponding transformation data types:

JSON Kafka Data Type	Transformation Data Type	Range and Description
BOOLEAN	Integer	TRUE (1) or FALSE (0)
DOUBLE	Double	Precision 15
INTEGER	Integer	-2,147,483,648 to 2,147,483,647 Precision of 10, scale of 0
LONG	Bigint	Precision of 19 digits, scale of 0
STRING	Nstring	1 to 104,857,600 characters

Avro Kafka Data Types and Transformation Data Types

Avro Kafka data types map to transformation data types that the PowerCenter Integration Service uses to move data across platforms.

The following table lists the Avro Kafka data types that the PowerCenter Integration Service supports and the corresponding transformation data types:

Avro Kafka Data Type	Transformation Data Type	Range and Description
BOOLEAN	Integer	TRUE (1) or FALSE (0)
DOUBLE	Double	Precision 15
FLOAT	Double	Precision 15
INT	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
LONG	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
STRING	NString	1 to 104,857,600 characters

INDEX

A

Avro Kafka data types
transformation data types [34](#)

C

configuration
overview [8](#)

F

filter [22](#)

I

Import
connection properties [15](#)
source definition [15](#)
target definition [15](#)

J

JSON Kafka data types
transformation data types [33](#)

K

Kafka
connections [26](#)
mappings [22](#)
overview [7](#)

Kafka (*continued*)

source [15](#)
target [15](#)

Kafka connections

properties [26](#)
connection configuration [27](#)

Kerberos Kafka

prerequisites [12](#)

M

mapping
filter [22](#)
source [22](#)
target [22](#)

P

plug-in registration
administrator tool [9](#)
PowerExchange for Kafka
configuration [8](#)
overview [7](#)
plug-in registration [8](#)

S

session
overview [25](#)
source properties [28](#)
target properties [29](#)
sessions
message recovery [29](#)