

How to Use Topic Patterns in Kafka Data Objects

Abstract

You can read from Kafka sources in streaming mappings. When you configure a Kafka data object to read from a Kafka topic, you can specify the topic name or use a regular expression for the topic name pattern. This article describes the how to specify a topic pattern and the guidelines to use for the topic pattern while creating the data objects.

Supported Versions

- Informatica Intelligent Streaming 10.2

Table of Contents

Overview.	2
Example.	2
Creating Kafka Data Objects.	2
Regular Expression Syntax Guidelines.	5
Converting COBOL Syntax to perl Compatible Regular Expression Syntax.	5
Converting SQL Syntax to perl Compatible Regular Expression Syntax.	6

Overview

You can read from and write to Kafka in streaming mappings that run on the Spark engine. Create a Kafka data object to read from or write to Kafka topics.

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

To read topic data from Kafka with Intelligent Streaming, you create a Kafka data object in Informatica Developer (the Developer tool). When you configure the Kafka data object, specify the topic name that you 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.

Example

You have captured change data from your sources and stored the changes in multiple topics in Kafka. You want to write the data to HDFS. You can subscribe to all topics that match the specified pattern to get dynamically assigned partitions. Specify a regular expression to subscribe to all topics that match the pattern. For example, if the Kafka topic names are `cdc_table1`, `cdc_table2`, and `cdc_table3`, you can specify the regular expression `cdc_.*` to subscribe to all topics.

Creating Kafka Data Objects

Create a Kafka data object in the Developer tool to add to streaming mappings.

1. Select a project or folder in the **Object Explorer** view in Developer tool.
2. Click **File > New > Data Object**.

3. Select **Kafka** and click **Next**.

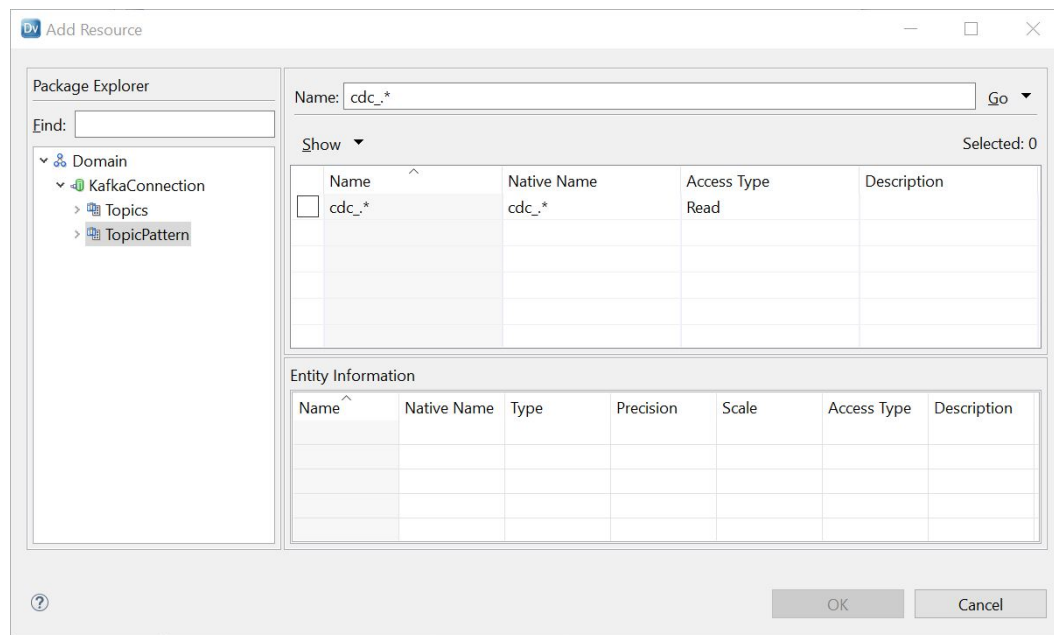
The data object dialog box appears.

4. Enter a name for the data object.
5. Click **Browse** next to the **Location** option and select the target project or folder.
6. Click **Browse** next to the **Connection** option and select the Kafka connection.
7. To add a resource, click **Add** next to the **Selected Resources** option.

The **Add Resource** dialog box appears.

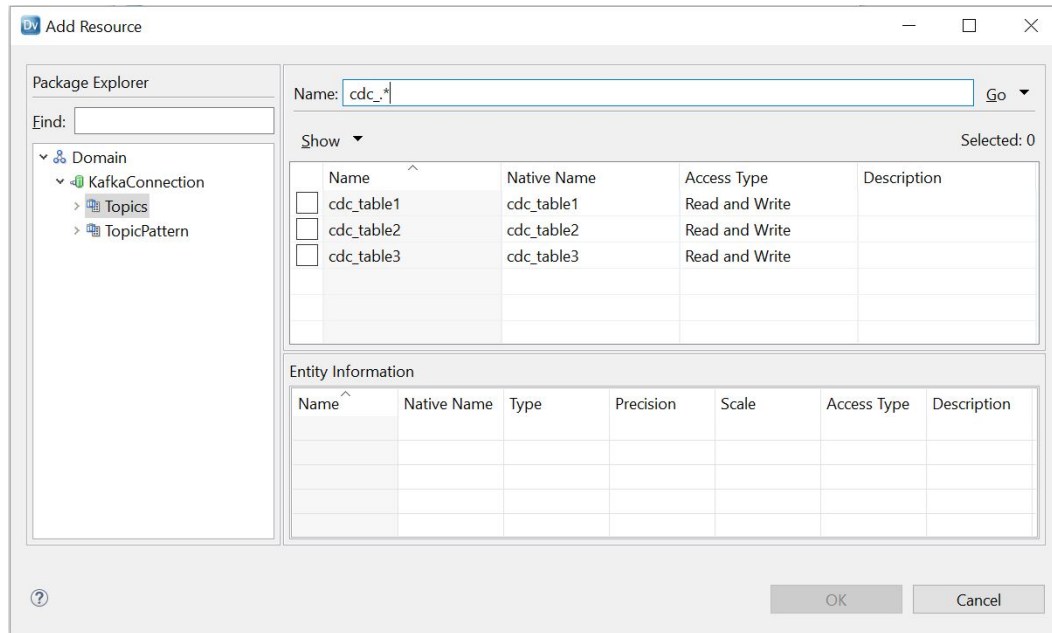
8. To specify a topic pattern for the topic name, in the **Add Resource** dialog box, select **TopicPattern**.
9. Specify a regular expression pattern for the topic name that you want to read from. Use the regular expression syntax guidelines to specify the pattern.

The following image shows the `cdc_.*` topic pattern:



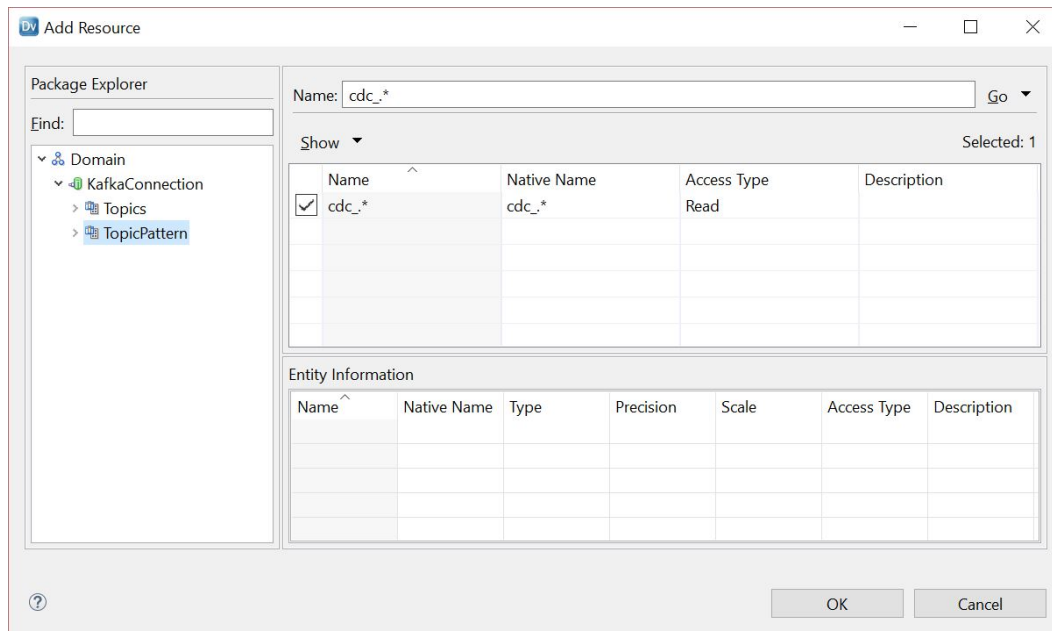
To verify the topics that match the `cdc_.*` topic pattern, select **Topic**.

The following image shows the topic names that match the `cdc_.*` pattern:



10. Select the topic pattern that you want to add and click **OK**.

The following image shows the `cdc_.*` topic pattern selected:



11. Click **Finish**.

The Kafka data object appears under Data Objects in the project or folder in the **Object Explorer** view. You can create a read data object operation to add in streaming mappings.

Regular Expression Syntax Guidelines

You must use perl compatible regular expression syntax when you create a regular expression.

The following table describes perl compatible regular expression syntax guidelines to create a regular expression:

Syntax	Description
.	(a period) Matches any one character.
[a-z]	Matches one instance of a character in lower case. For example, [a-z] matches ab. Use [A-Z] to match characters in upper case.
\d	Matches one instance of any digit from 0-9.
\s	Matches a whitespace character.
\w	Matches one alphanumeric character, including underscore (_)
()	Groups an expression. For example, the parentheses in (\d-\d\d\d) groups the expression \d\d-\d\d, which finds any two numbers followed by a hyphen and any two numbers, as in 12-34.
{}	Matches the number of characters. For example, \d{3} matches any three numbers, such as 650 or 510. Or, [a-z]{2} matches any two letters, such as CA or NY.
?	Matches the preceding character or group of characters zero or one time. For example, \d{3}(-d{4})? matches any three numbers, which can be followed by a hyphen and any four numbers.
*	(an asterisk) Matches zero or more instances of the values that follow the asterisk. For example, *0 is any value that precedes a 0.
+	Matches one or more instances of the values that follow the plus sign. For example, \w+ is any value that follows an alphanumeric character.

Converting COBOL Syntax to perl Compatible Regular Expression Syntax

If you are familiar with COBOL syntax, use the following information to help you write perl compatible regular expressions.

The following table describes examples of COBOL syntax and their perl compatible equivalents:

COBOL Syntax	perl Syntax	Description
9	\d	Matches one instance of any digit from 0-9.
9999	\d\d\d\d or \d{4}	Matches any four digits from 0-9, as in 1234 or 5936.

COBOL Syntax	perl Syntax	Description
x	[a-z]	Matches one instance of a letter.
9xx9	\d[a-z][a-z]\d	Matches any number followed by two letters and another number, as in 1ab2.

Converting SQL Syntax to perl Compatible Regular Expression Syntax

If you are familiar with SQL syntax, use the following information to help you write perl compatible regular expressions.

The following table describes examples of SQL syntax and their perl compatible equivalents:

SQL Syntax	perl Syntax	Description
%	. *	Matches any string.
A%	A.*	Matches the letter "A" followed by any string, as in Area.
_	. (a period)	Matches any one character.
A_	A.	Matches "A" followed by any one character, such as AZ.

Authors

Vidya Vasudevan

Neha Raj