Generating a UUID Using a Java Transformation in PowerCenter
Abstract

A Universal Unique Identifier (UUID) uniquely identifies a record. This article describes how you can use a Java transformation in a PowerCenter mapping to generate a UUID and send it to the target system.

Supported Versions

- PowerCenter 8.6.1 HotFix13 - 9.1.0

Table of Contents

Overview .....................................................................  2
Example.....................................................................  2
Java Transformation .............................................................  2
Configuring an Output Port .........................................................  3
Configuring the Java Code Tab. .......................................................  3
Import Packages Tab ...........................................................  3
On Input Row Tab. .............................................................  3

Overview

A UUID is a 128-bit number and is represented by 32 hexadecimal alphanumeric characters. You do not require a centralized authority to administer a UUID. Therefore, you can completely automate the generation of a UUID and use it for multiple purposes. Here is an example of a UUID: e5dabe13-ff37-423b-b980-538e7132cd67.

Example

Your organization needs to integrate data from different databases, and you need to generate unique IDs that are unique across systems. A regular ID is unique within the table of the database. However, there may be a conflict of ID numbers when you integrate data from different databases. If the IDs are integers, an external entity can decipher the number of records and size of the database. To generate unique, non numeric IDs, use a UUID that is unique across systems and which you can generate using an external application.

You can use the Java transformation to define simple or moderately complex transformation functionality without advanced knowledge of the Java programming language or an external Java development environment.

You can use a passive Java transformation in a mapping to generate a UUID and send it to the target system.

Java Transformation

Use the Java transformation to read each row of data and pass the data to output ports. Define the Java transformation functionality on the Java Code tab.

To generate a UUID, complete the following steps:

1. Configure an output port
2. Configure the Java Code tab
Configuring an Output Port

Configure an output port to add the UUID. Use the port name as a variable in the Java code snippet.

In the Ports view, create an output port called UUID_Out of type String and of length 36.

Configuring the Java Code Tab

The Java Code tab contains multiple code entry tabs. Use the code entry tabs to enter Java code snippets that define the Java transformation functionality. This example uses the Import Packages tab and the On Input Row tab.

Import Packages Tab

You can use the Import Packages tab to import third-party, built-in, and custom Java packages for an active or passive Java transformation. After you import the packages, you can use them on the other tabs in the Java Code tab.

Use the Import Packages tab of the Java Code to import a Java package named java.util.UUID. The java.util.UUID package is supported from Java 1.5 onwards.

Enter the following code on the Import Packages tab:

```java
import java.util.UUID
```

On Input Row Tab

You can use the On Input Row tab to define the behavior of the transformation when it receives an input row.

Use the output port name you configured as a variable in the following Java code snippet.

Enter the following code on the On Input Row tab:

```java
UUID uuid = UUID.randomUUID();
UUID_Out = uuid.toString();
```

Author

Maria Kurien Abraham  
Senior Technical Writer

Acknowledgements

The author would like to acknowledge Sidhartha Sarat Bardoloye, QA Manager, for his technical assistance.