How-To Library

Informatica

Reading a JSON File Using an Amazon S3 V2 Connector

Abstract

You can use Amazon S3 V2 Connector to read a JSON file through a binary port. To read a JSON file, you must use the Hierarchy Parser and Java transformations in the Amazon S3 V2 mapping. Adding the transformations enable the Secure Agent to convert the data of the binary format to the string format. This article explains how to create a mapping to read a JSON file using Amazon S3 V2 Connector.

Supported Versions

- Informatica® Cloud Data Integration Amazon S3 V2

Table of Contents

Overview. 2
Prerequisite Tasks. 2
Creating an Amazon S3 V2 Mapping to Read a JSON File. 2

Overview

When you create an Amazon S3 V2 mapping, use the Hierarchy Parser and Java transformations to read JSON files. You cannot create a mapping to write JSON files. However, in elastic mappings, you can directly read and write JSON files.

Prerequisite Tasks

Before you use Amazon S3 V2 Connector to read a JSON file through the binary port, you must complete the following prerequisite tasks:

1. Verify whether you have the `AgentCompiler` custom license to use the Java transformation.
2. Verify whether you have the following custom licenses to use the Hierarchy Parser transformation:
    - DataTransformation
    - UDTforHierarchy
    - saas-xmetadataread
3. Set the value of the `JDK_HOME` environment variable to the Java installation path in the Secure Agent machine.

Creating an Amazon S3 V2 Mapping to Read a JSON File

Add the Hierarchy Parser and Java transformations in an Amazon S3 mapping to read a JSON file.

1. In **Data Integration**, click **New** > **Mappings** > **Create** .

    The **New Mapping** dialog box appears.
2. Enter a name, location, and description for the mapping.
3. On the Source transformation, specify a name and description in the general properties.
4. On the **Source** tab, perform the following steps to provide the source details to read data from an Amazon S3 source:
    1. In the **Connection** field, select the **Amazon S3 V2** connection.

    2. In the **Source Type** field, select **Single** object.
    3. In the **Object** field, select the source object of the JSON file.

       Note: By default, the format type of a JSON file is binary.
    4. In the **Formatting Options** field, set the value to **None**.
5. Add a Java transformation.
6. Click **Java**.

   The **Java Editor** tab appears.
7. Click **Inputs**.

   The **Inputs** tab appears.
8. Add the following code to import the required Java packages in the **Import Packages** section:

   ```
   import java.nio.charset.StandardCharsets;
   ```
9. Add the following code to convert the data of binary format into string format in the **On Input Row** section:

   ```
   <variable_name>= new String(data, StandardCharsets.UTF_8);
   generateRow();
   ```

   For example,

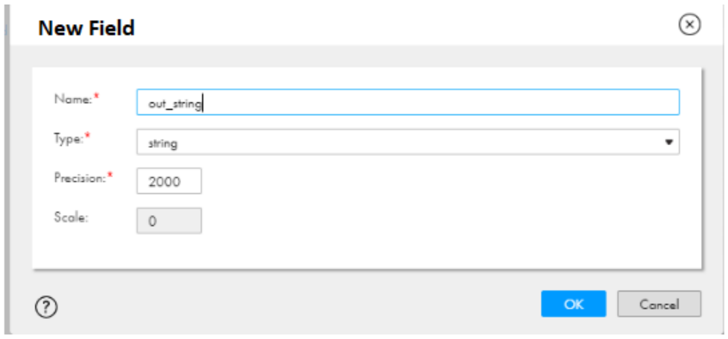
   ```
   out_string= new String(data, StandardCharsets.UTF_8);
   generateRow();
   ```
10. Click **Outputs**.

    The **Outputs** tab appears.
11. Click **+ Create a New Field**.

    The **New Field** dialog box appears.
12. In the **New Field** dialog box, define the variable names that you added in the **On Input Row** section.

    For example, define the `out_string` variable name.

    The following image shows an example of the **New Field** dialog box:



New Field

Name:* out_string

Type:* string

Precision:* 2000

Scale: 0

13. Click **OK**.
14. In the **Runtime Environment** field, select the Secure Agent that you want to use and click **Compile**.

    The compilation successful message is displayed.
15. Add a Hierarchy Parser transformation.
16. Map the input ports from the Java transformation to the Hierarchy Parser transformation.
17. Click on the Hierarchy Parser transformation.
18. In the **Incoming Fields** tab, perform the following steps:
    1. Click the **+** add icon and select **Delete Rule** to delete any existing rules.

    2. Click the **+** add icon to add a new rule.
    3. Click **Configure**.
       The **Configure Field Rules** dialog box appears.
    4. In the **Rule Details** tab, select only the output strings.
       For example, **out_string**.
    5. Click **OK**.

19. In the **Input Settings** tab, select the **Input Type** as **Buffer** and select an existing or create a new hierarchical schema.
20. Map the input ports from the Java transformation to the Hierarchy Parser transformation.
21. In the **Input Field Selection** tab, map the incoming fields to the hierarchical schema input fields.
22. In the **Field Mapping** tab, expand the element name and click **Map all descendants**.
23. Map the Hierarchy Parser transformation to the target.

    The **Select Output Group** dialog box appears.
24. Select the required output group and click **OK**.
25. Click **Save** > **Run** to run the mapping.

Author

Subhashree Salam

Acknowledgements

The author would like to acknowledge Aditi Tellar Nayak, for her technical assistance with this article.