CDC Software Platform Connector

Table of Contents

Overview

The Anypoint CDC Software Connector provides connectivity to the CDC Software API, enabling you to interface with CDC Software to retrieve telephony events and perform operations on more than a dozen telephony systems.

Read through this user guide to understand how to set up and configure a basic flow using the connector.

Important Concepts

This document assumes that you are familiar with Mule, Anypoint Connectors, and Anypoint Studio.

Hardware and Software Requirements

To use the CDC Software Platform Connector you must have in-place an installation of the CDC Platform. For additional system support information, visit the CDC Software Platform Connector Release Notes.

Application/Service	Version
CDC Platform	1.0 or later.
Mule Runtime	3.6.0 or later.
Asterisk and Asterisk based systems	FreePBX and similar
Broadsoft	eXtended Serveices Interface (XSI)
Alcatel-Lucent OMNI	CSTA Compliant Versions
Avaya Aura	Application Enablement Services 5.0 or above
Avaya Aura	Avaya Communications Control Toolkit
Avaya IP Office	CTI Link Pro 8.x and above
Cisco CUCM	7.5 and above
Cisco UCCE	7.5 and above
Cisco UCCX	7.5 and above
InContact	Webservices API
Mitel	Open Integration Gateway 3.0 and above
Mitel	TAPI Compatible
ShoreTel	Webservices API
Switchvox	Triggers over HTTP
Vonage Business	HTTP API
Other systems	TAPI, CSTA, or TSAPI

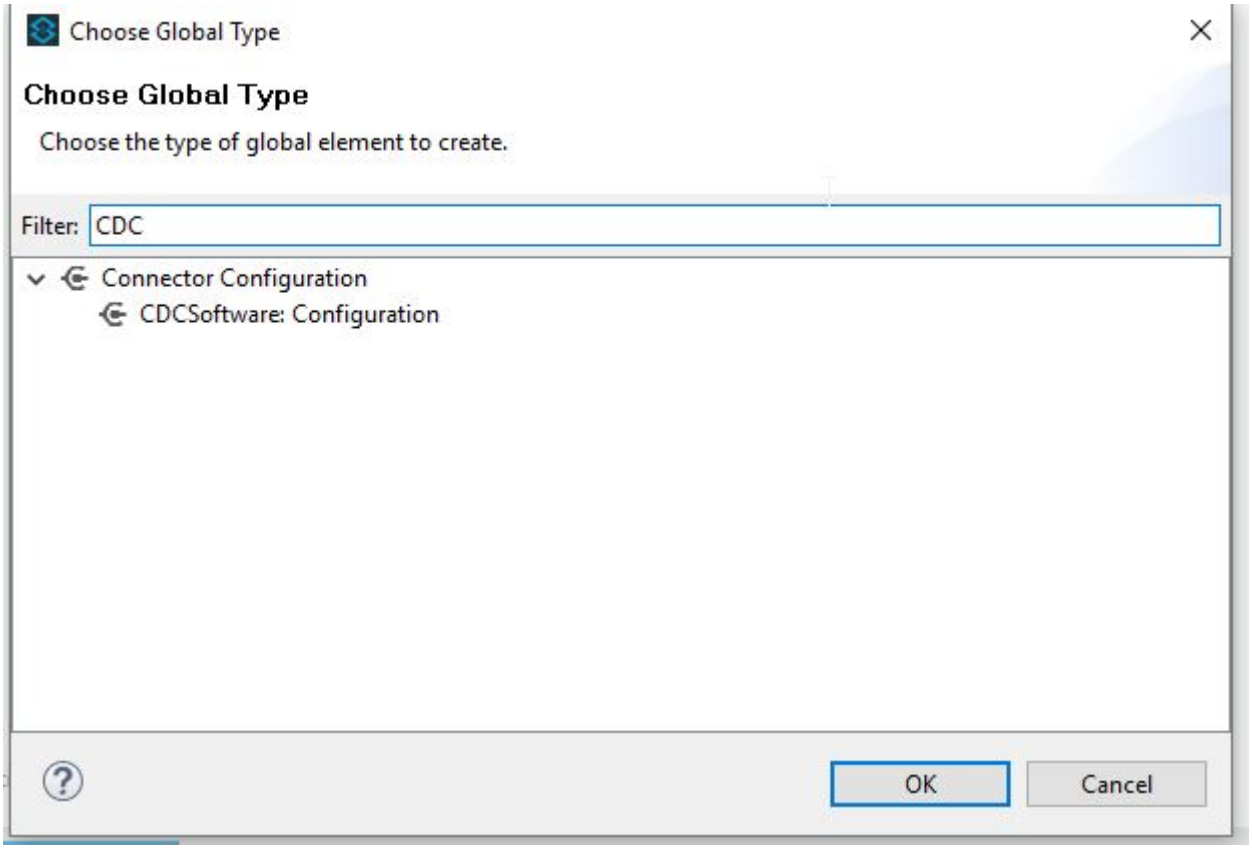
How to Install

You can install the connector in Anypoint Studio using the instructions in Installing a Connector from Anypoint Exchange.

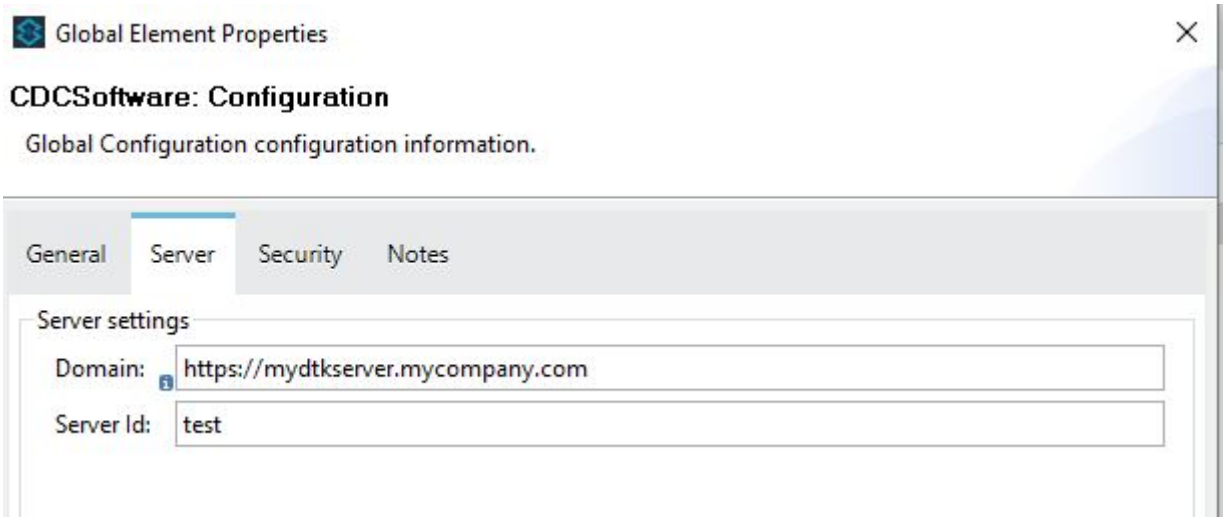
Configuring the Global Element

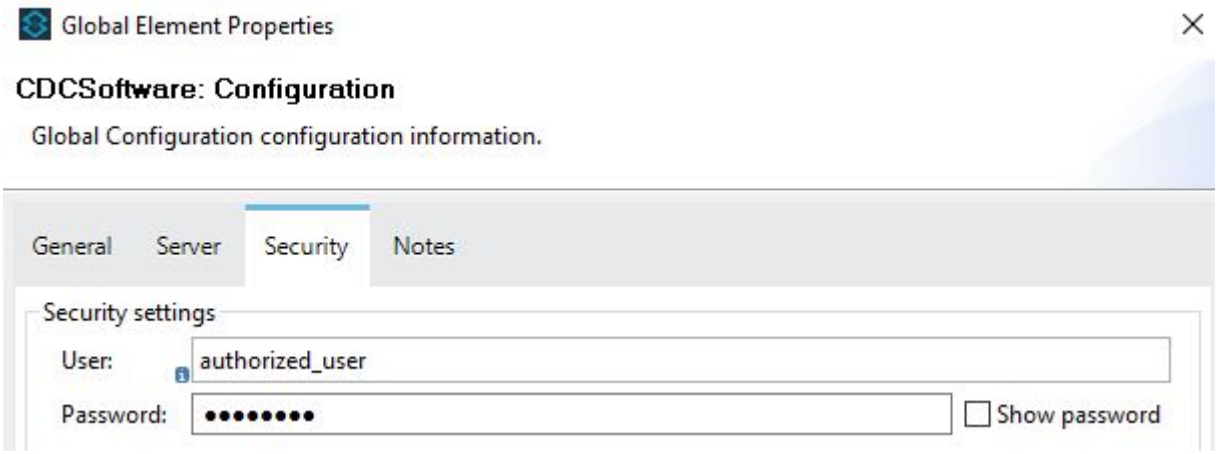
To use the CDC Software connector in your Mule application, you must configure a global CDC Software element that can be used by all the CDC Software connectors in the application (read more about Global Elements.)

- Click the Global Elements tab at the base of the canvas, then click Create.
- In the Choose Global Type window, expand Connector Configuration, and click CDC Software: Configuration.



- Click Ok
- Enter the global element properties:





Parameter	Description
Name	Enter a name for the configuration to reference it.
Domain	The URL of your DTK installation.
Server Id	The instance name of your installation.
User	The authorized user.
Password	The authorized user password.

You can either enter your credentials into the global configuration properties, or reference a configuration file that contains these values. For simpler maintenance and better re-usability of your project, Mule recommends that you use a configuration file. Keeping these values in a separate file is useful if you need to deploy to different environments, such as production, development, and QA, where your access credentials differ. See Deploying to Multiple Environments for instructions on how to manage this.

- Click OK to save the global connector configurations.

Upgrading from an Older Version

If you're currently using an older version of the connector, a small popup appears in the bottom right corner of Anypoint Studio with an "Updates Available" message.

1. Click the popup and check for available updates.

2. Click the Connector version checkbox, click **Next**, and follow the instructions provided by the user interface.
3. **Restart** Studio when prompted.
4. After restarting, when creating a flow and using the connector, if you have several versions of the connector installed, you may be asked which version you would like to use. Choose the version you would like to use.

Additionally, we recommend that you keep Studio up to date with its latest version.

Operations

CDC Software connector as a stream connector currently supports the following list of operations:

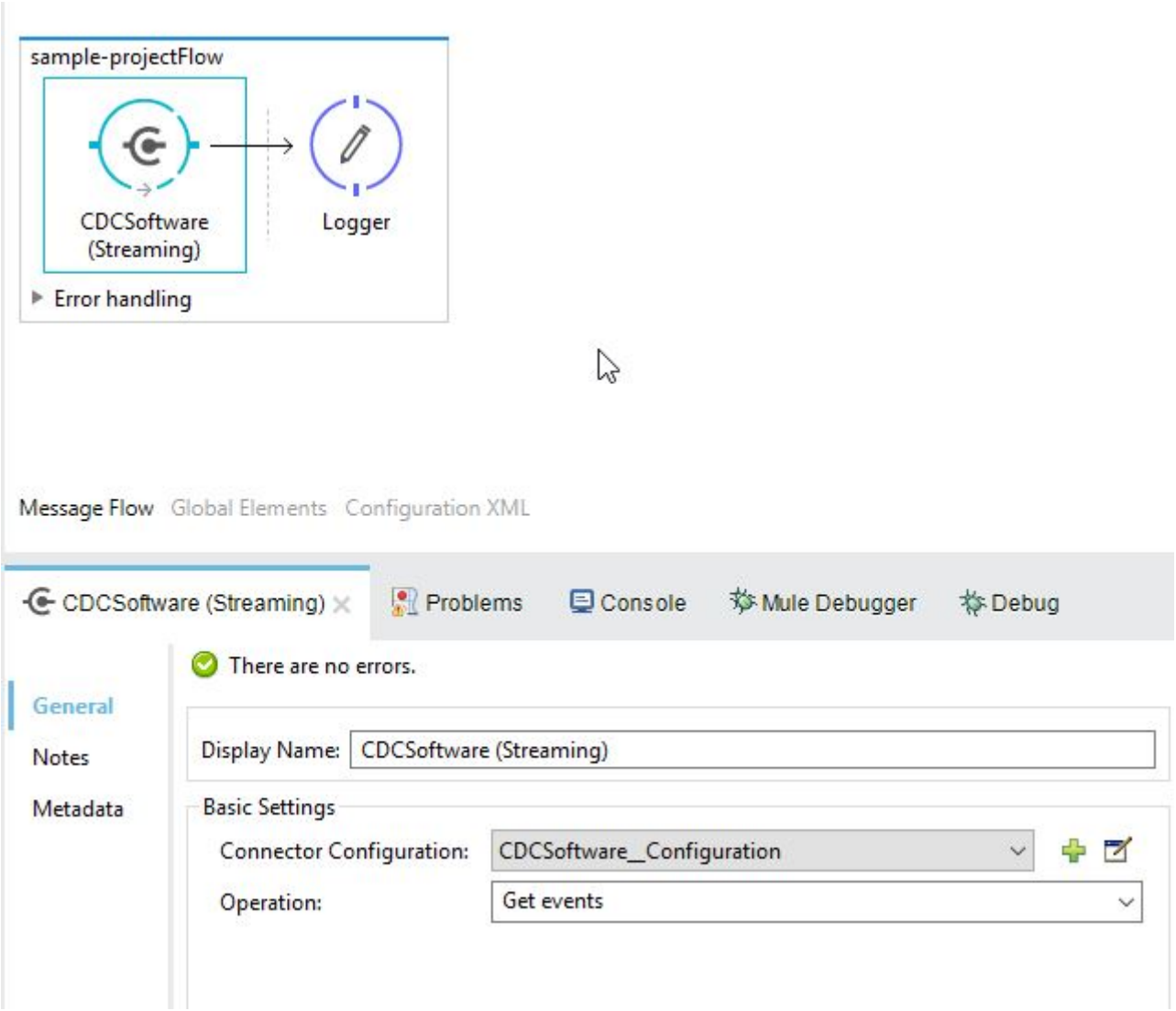
- Get Events

CDC Software connector as an operation connector currently supports the following list of operations:

- Make Call
- Answer Call
- Release Call

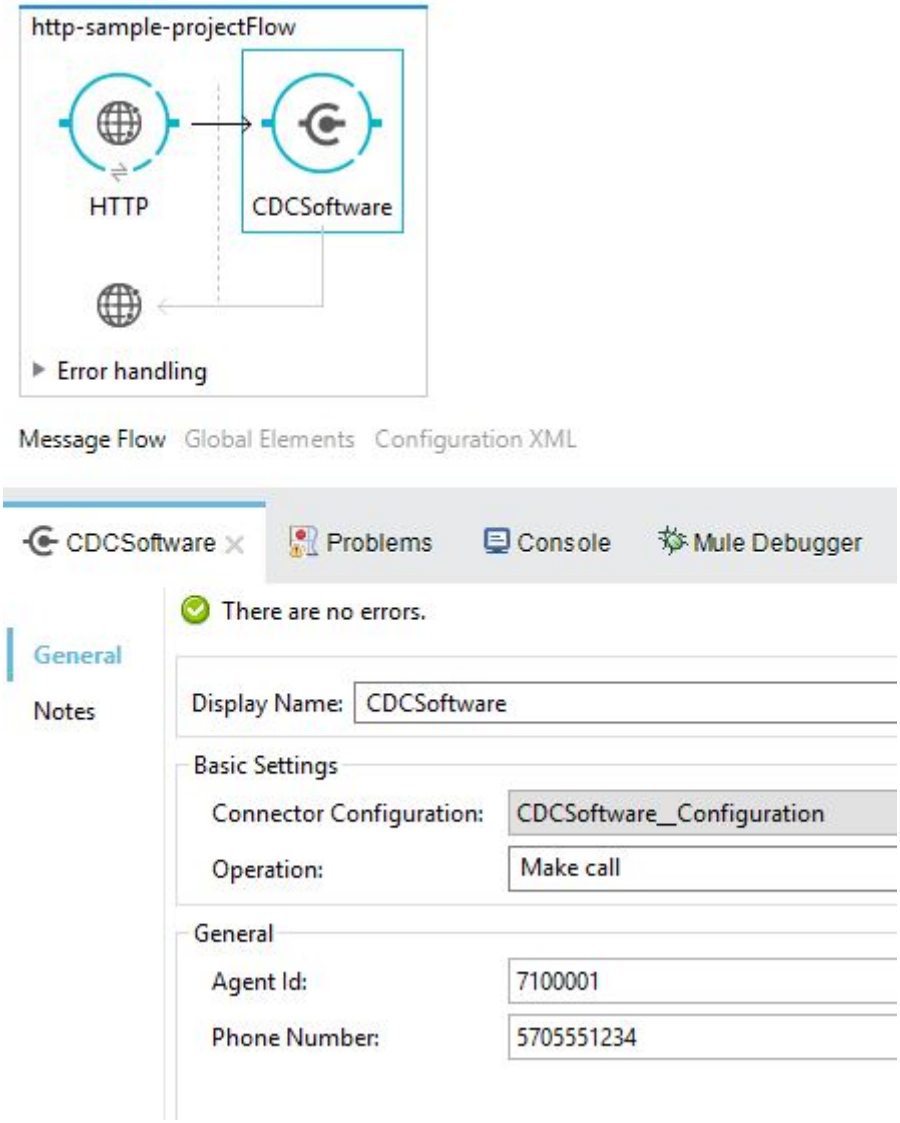
Adding Connector to Mule Flow as a Stream

1. Create a new Mule project in Anypoint Studio.
2. Drag the CDC Software connector onto the canvas, then select it to open the properties editor.
3. Configure the connector by using previously define configuration and selecting the “Get events” operation.



Adding Connector to Mule Flow as a Process

1. Create a new Mule project in Anypoint Studio.
2. Drag the CDC Software connector onto the process section of the flow.
3. Configure the connector by using previously define configuration and select the processor operation.



Common Use Cases

The primary purpose of the CDC Software Platform Connector is to expose the primary call events of the supported telephony systems directly to Mulesoft to support the triggering of actions within other Mulesoft connectors. For example, the answering of a call could be mapped to a action (create, modify, search etc.) in another Mulesoft connector. Examples might be:

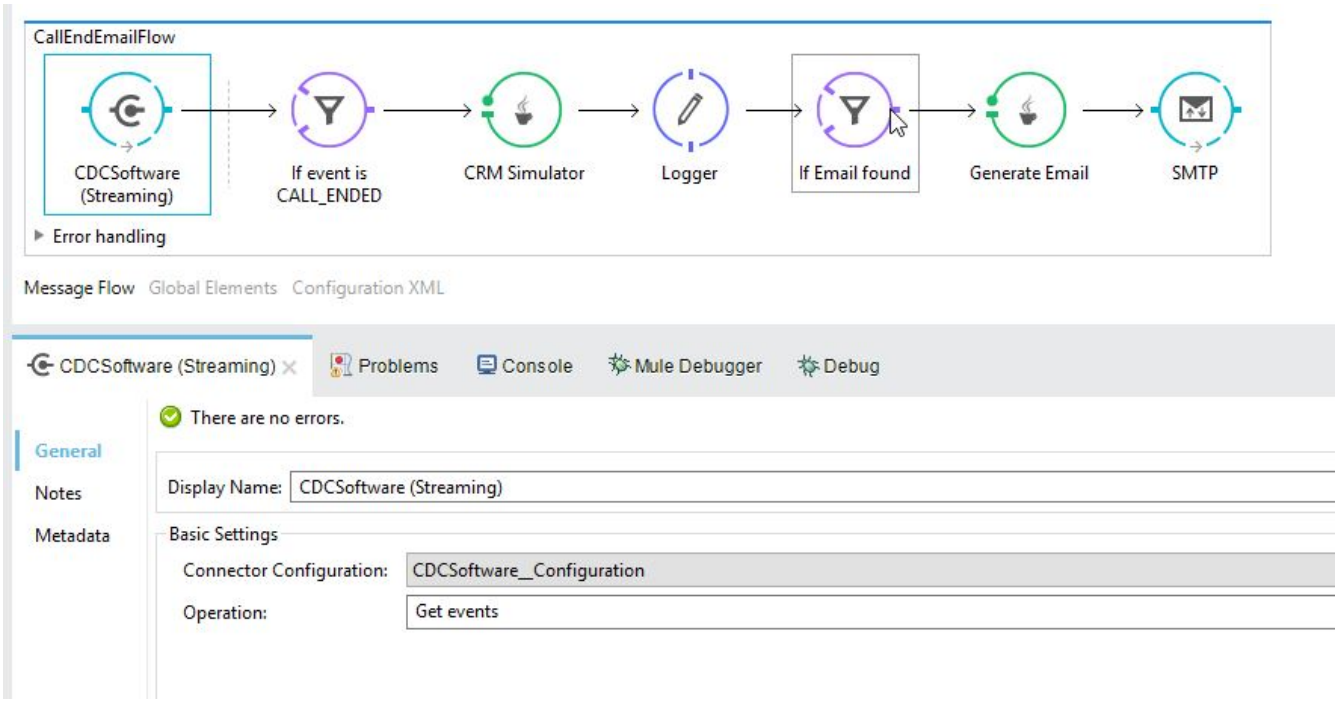
- Screen-popping a CRM.
- Updating or opening a webpage or web service
- Sending an email
- Writing or reading a record to/from a DB

See following use case examples:

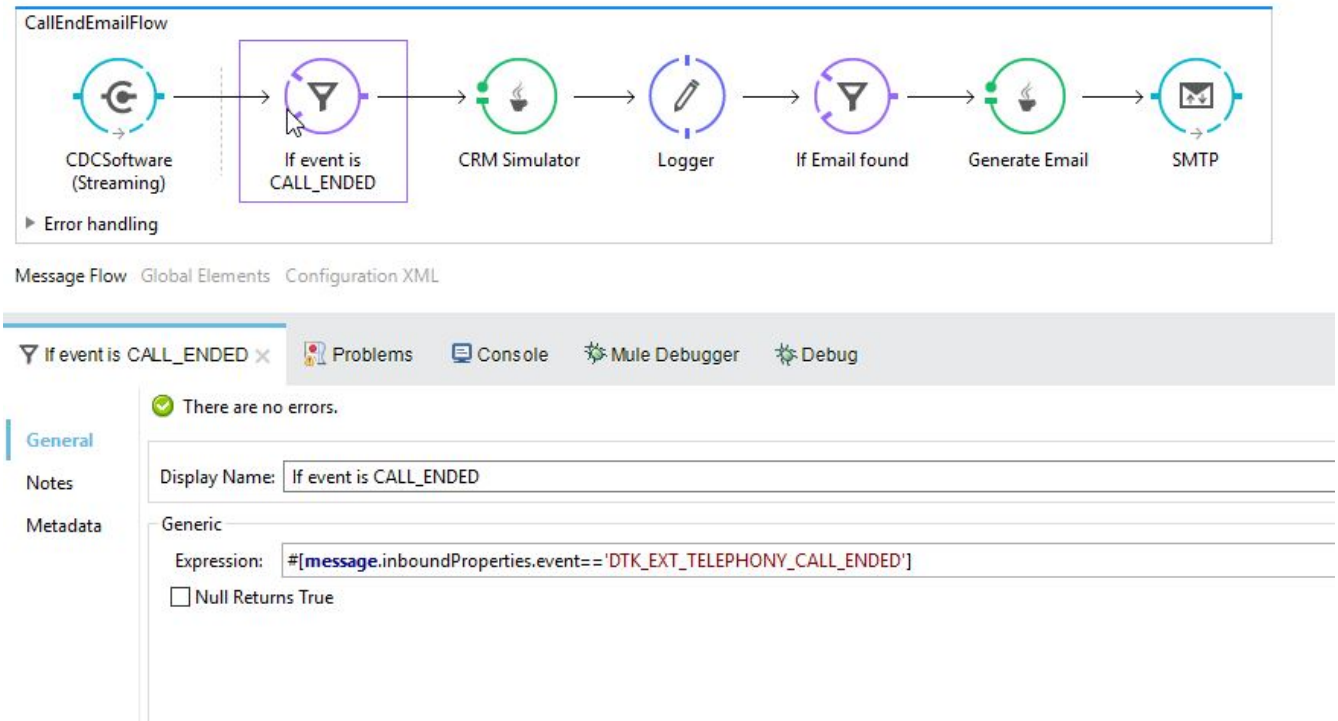
- Trigger email via phone call ...

Trigger email

Create a Mule application that generates an email at the end of a call.



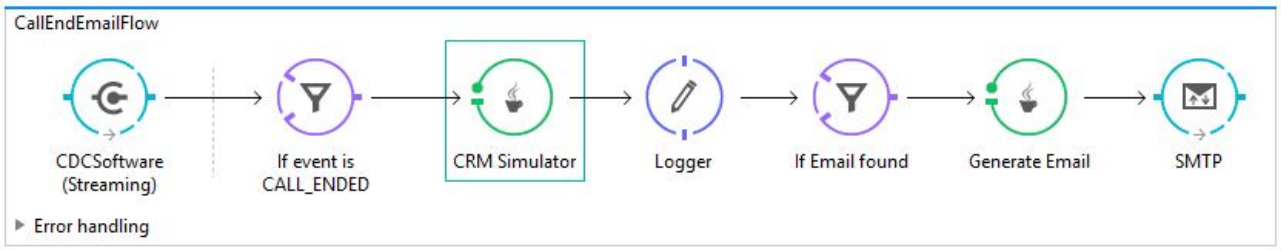
- Create a new Mule project in Anypoint Studio.
- Drag CDCSoftware connector on to the canvas, then select it to open the properties editor console.
- Configure the CDCSoftware connector with domain, server id, user, and password.
- Drag an Expression Filter, and double-click the connector to open its Properties Editor.



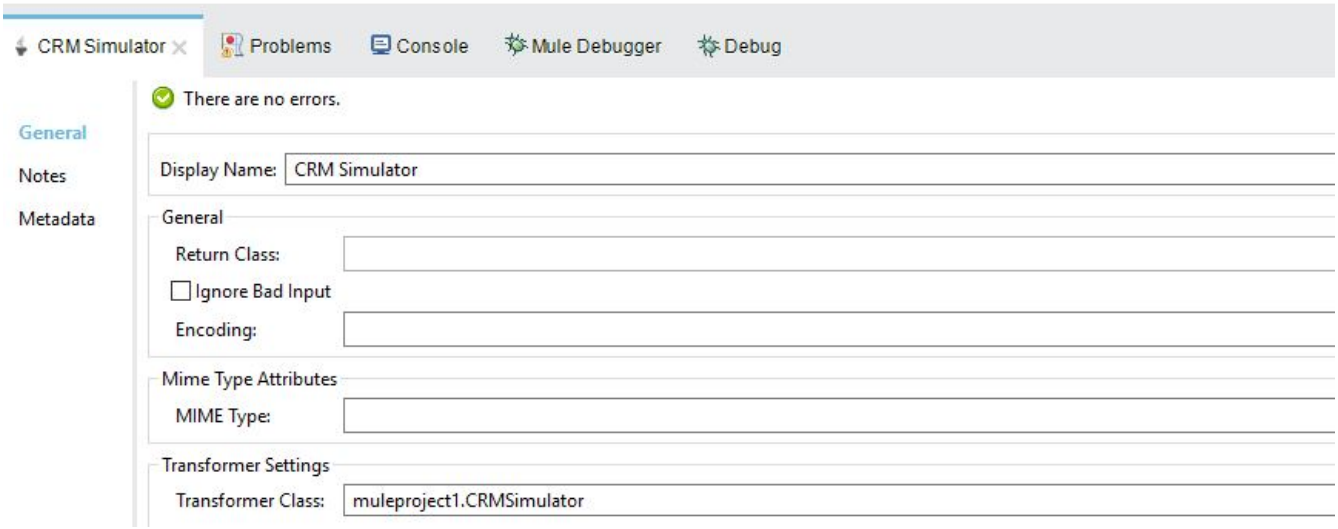
- Edit the expression to filter by DTK event CALL END

#[message.inboundProperties.event=='DTK_EXT_TELEPHONY_CALL_ENDED']

- Drag a Java Transformer to create your custom logic to retrieve customer email by phone as example below:



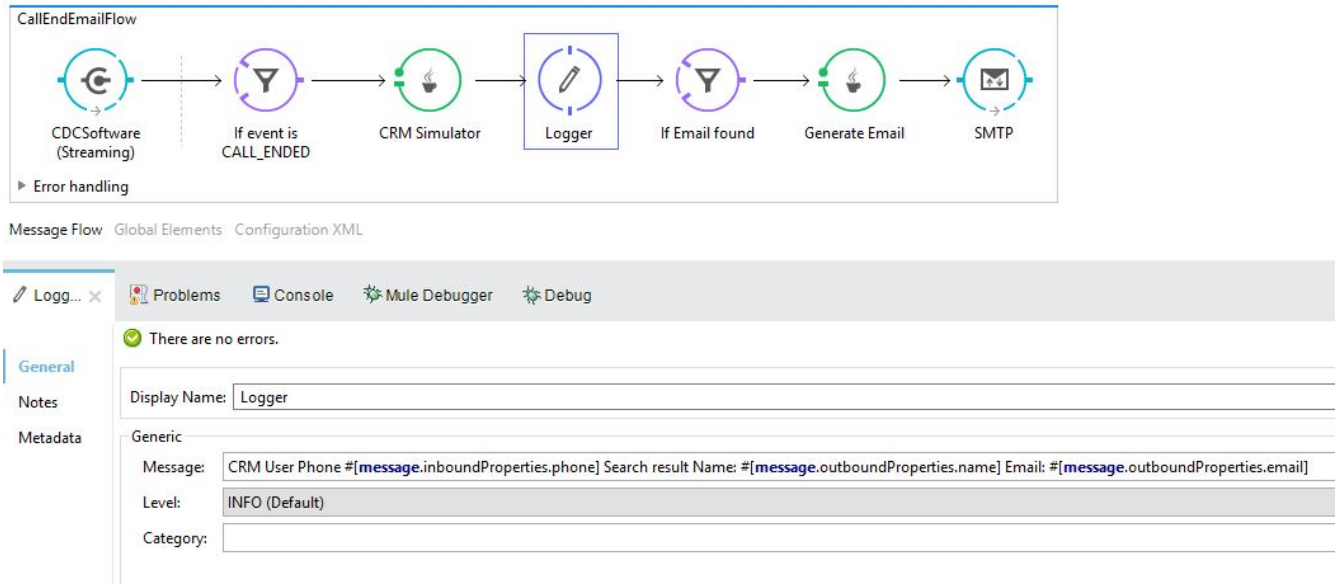
Message Flow Global Elements Configuration XML



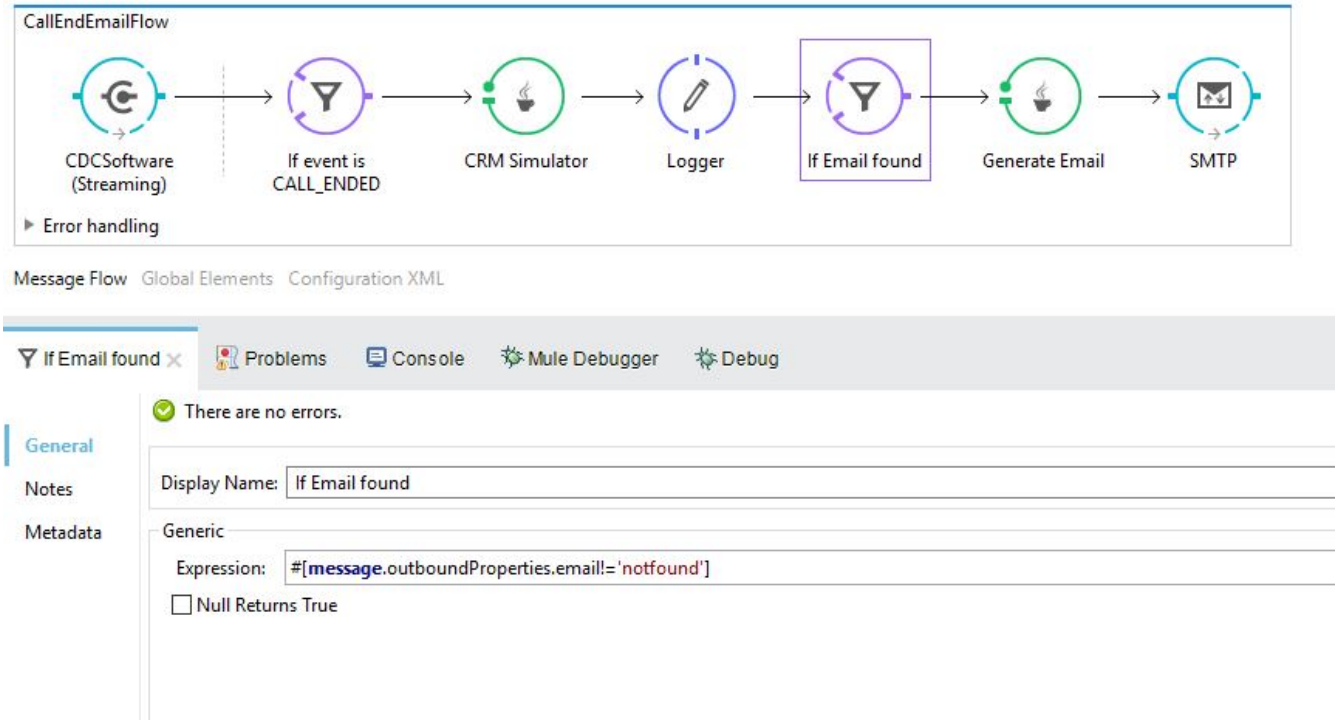
- Example Java code

```
1.  package muleproject1;
2.
3.  import java.util.Map;
4.  import java.util.HashMap;
5.  import org.mule.api.MuleMessage;
6.  import org.mule.api.transformer.TransformerException;
7.  import org.mule.api.transport.PropertyScope;
8.  import org.mule.transformer.AbstractMessageTransformer;
9.  public class CRMSimulator extends AbstractMessageTransformer{
10. public Object transformMessage(MuleMessage message,
11. String outputEncoding) throws TransformerException {
12.
13. String phone =
14. message.getProperty("phone",PropertyScope.INBOUND).toString();
15.         Map<String,Object> map = new HashMap<String,Object>();
16.
17.         switch (phone) {
18.             case "3055551234":
19.                 map.put("name", "joe");
20.                 map.put("email", "joe@mycompany.com");
21.                 break;
22.
23.
24.             default:
25.                 map.put("name", "notfound");
26.                 map.put("email", "notfound");
27.
28.                 break;
29.         }
30.
31.         message.addProperties(map, PropertyScope.OUTBOUND);
32.         return message;
33.     }
34. }
```

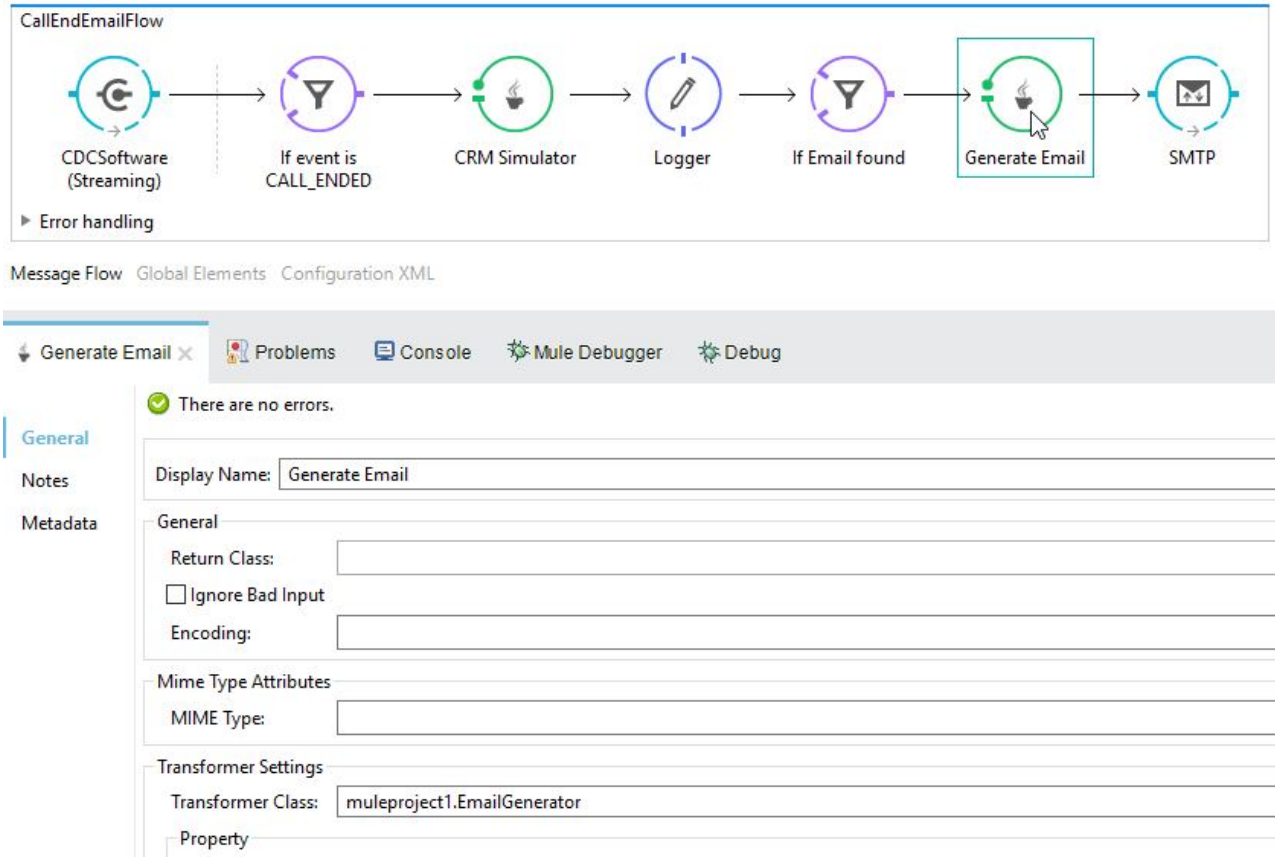
- Drag a Logger component to log the results of your Java code



- Drag an expression filter to filter message that do not contain ‘notfound’ in the email property #[message.outboundProperties.email!='notfound']



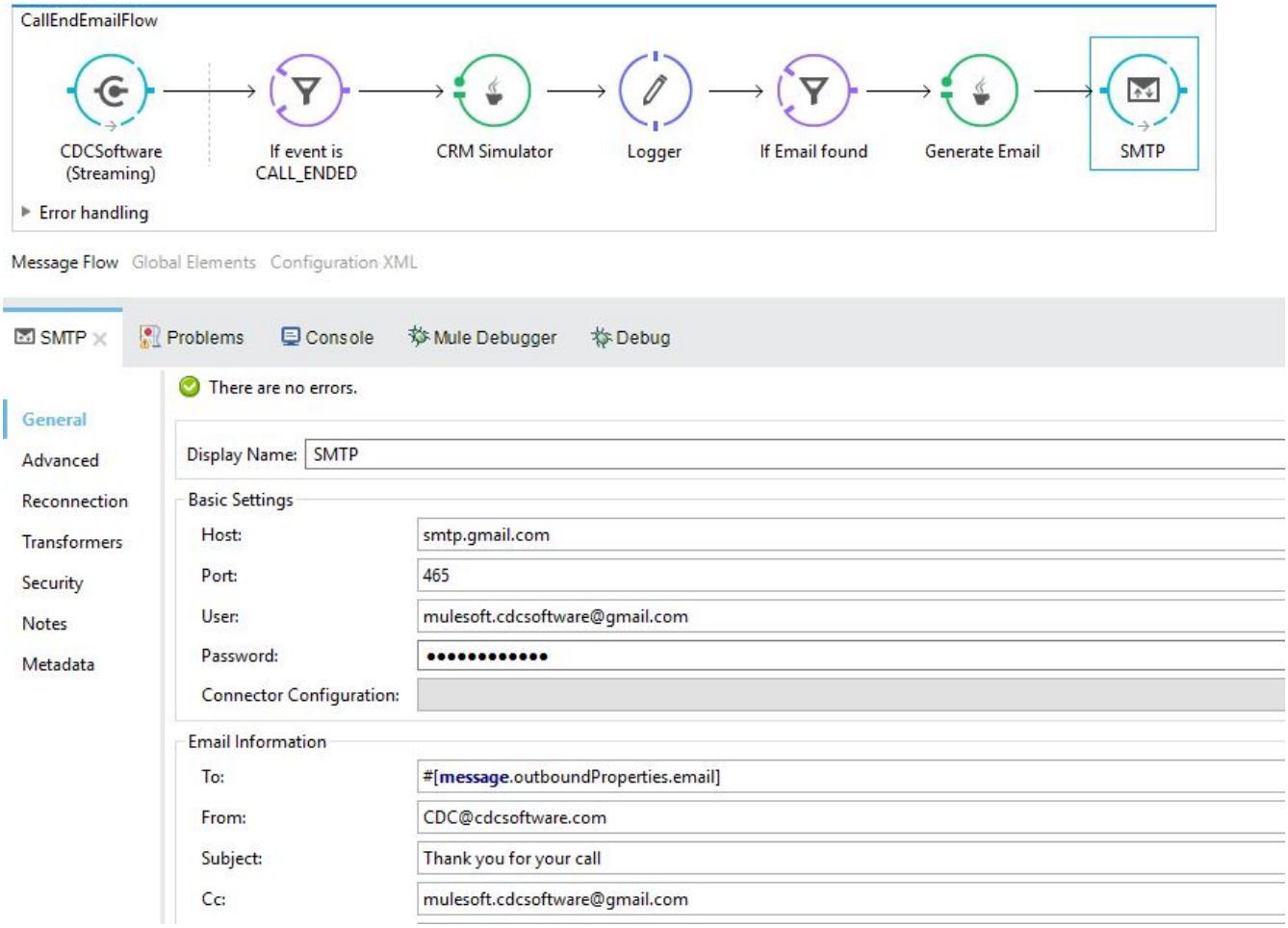
- Drag another Java Transformer to customize the email body .



- Example Java code

```
1.  package muleproject1;

2.  import org.mule.api.MuleMessage;
3.  import org.mule.api.transformer.TransformerException;
4.  import org.mule.api.transport.PropertyScope;
5.  import org.mule.transformer.AbstractMessageTransformer;

6.  public class EmailGenerator extends AbstractMessageTransformer {

7.  public Object transformMessage(MuleMessage message, String

8.  outputEncoding) throws TransformerException {

9.  String phone =message.getProperty("phone",PropertyScope.INBOUND).toString();
10.
11. String name =message.getProperty("name",PropertyScope.OUTBOUND).toString();
12.
13. StringBuilder sb = new StringBuilder();
14. sb.append("Dear: ");
15. sb.append(name);
16. sb.append("\r\n");
17. sb.append("We would like to thank you for your call from ");
18. sb.append(phone);
19.
20. return sb.toString();
21.
22. }
23. }
```

- Drag an SMTP endpoint to send the email to the customer that was found previously as shown in following image.



Resources

- Access the CDC Software Platform Connector Release Notes.