Tim Chapin

support@cdc-tek.com

CDC Software Platform Connector for Mulesoft AnyConnect



Contents

[Introduction 2](#_Toc481838776)

[Prerequisites 3](#_Toc481838777)

[Compatibility 3](#_Toc481838778)

[Installation 4](#_Toc481838779)

[Configure Global Element 4](#_Toc481838780)

[Using The Connector 6](#_Toc481838781)

[Adding the Connector to a Mule Flow as Stream Source 6](#_Toc481838782)

[Adding the Connector to a Mule Flow as a Process 7](#_Toc481838783)

[Use Cases 8](#_Toc481838784)

[Create a Mule application that generates an email at the end of a call 9](#_Toc481838785)

# Introduction

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.

# Prerequisites

To use the CDC Software connector, you must have the following:

* CDC Software Platform installed and configured for the telephony system(s) to be used.

# Compatibility

CDC Software connector is compatible with:

| **Application/Service** | **Version** |
| --- | --- |
| Mule Runtime | 3.5.0 or later |
| CDC Software Platform | 1.0 or above |
| Telephony Systems Supported:* Asterisk and Asterisk based systems.
* Broadsoft and Broadsoft based systems.
* Alcatel-Lucent OMNI Systems with CSTA
* Avaya Aura with Application Enablement Services 5.0 or above
* Avaya Aura with Avaya Communications Control Toolkit
* Avaya IP Office 8.x and above with CTI Link Pro
* Cisco CUCM 7.5 and above
* Cisco UCCE 7.5 and above
* Cisco UCCX 7.5 and above
* InContact
* Mitel with Open Integration Gateway 3.0 and above
* ShoreTel
* Switchvox
* Vonage Business
* Other systems that support: TAPI, CSTA, or TSAPI
 |  |

# Installation

The following sections describe how to install and configure this connector.

## Configure 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](https://docs.mulesoft.com/mule-user-guide/v/3.7/global-elements).)

* [**Studio Visual Editor**](https://docs.mulesoft.com/mule-user-guide/v/3.7/amazon-s3-connector#tab-1_1)
* [XML Editor](https://docs.mulesoft.com/mule-user-guide/v/3.7/amazon-s3-connector#tab-1_2)
1. Click the **Global Elements** tab at the base of the canvas, then click **Create**.
2. In the **Choose Global Type** window, expand **Connector Configuration**, and click **CDC Software: Configuration**.



1. Click **Ok**
2. 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](https://docs.mulesoft.com/mule-user-guide/v/3.7/deploying-to-multiple-environments) for instructions on how to manage this.

1. Click **OK** to save the global connector configurations.

# Using The Connector

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 the Connector to a Mule Flow as Stream Source

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 the Connector to a 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.



## 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

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



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



1. Edit the expression to filter by DTK event CALL END #[**message**.inboundProperties.event=='DTK\_EXT\_TELEPHONY\_CALL\_ENDED']
2. Drag a Java Transformer to create your custom logic to retrieve customer email by phone as example below:



1. Example Java code
2. **package** muleproject1;
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. String phone =
13. message.getProperty("phone",PropertyScope.***INBOUND***).toString();
14. Map<String,Object> map = **new** HashMap<String,Object>();
15.
16. **switch** (phone) {
17. **case** "3055551234":
18. map.put("name", "joe");
19. map.put("email", "joe@mycompany.com");
20. **break**;
21.
22.
23. **default**:
24. map.put("name", "notfound");
25. map.put("email", "notfound");
26.
27. **break**;
28. }
29.
30. message.addProperties(map, PropertyScope.***OUTBOUND***);
31. **return** message;
32. }
33. }
34. Drag a Logger component to log the results of your Java code



1. Drag an expression filter to filter message that do not contain ‘notfound’ in the email property

#[**message**.outboundProperties.email!='notfound']

1. Drag another Java Transformer to customize the email body



1. Example Java code
2. **package** muleproject1;
3. **import** org.mule.api.MuleMessage;
4. **import** org.mule.api.transformer.TransformerException;
5. **import** org.mule.api.transport.PropertyScope;
6. **import** org.mule.transformer.AbstractMessageTransformer;
7. **public** **class** EmailGenerator **extends** AbstractMessageTransformer{
8. **public** Object transformMessage(MuleMessage message, String
9. outputEncoding) **throws** TransformerException {
10. String phone =message.getProperty("phone",PropertyScope.***INBOUND***).toString();
11. String name =message.getProperty("name",PropertyScope.***OUTBOUND***).toString();
12. StringBuilder sb = **new** StringBuilder();
13. sb.append("Dear: ");
14. sb.append(name);
15. sb.append("\r\n");
16. sb.append("We would like to thank you for your call from ");
17. sb.append(phone);
18. **return** sb.toString();
19. }
20. }
21. Drag an SMTP endpoint to send the email to the customer that was found previously:

