

**MuleSoft Logging Standards**

**(Audit & Exceptions)**

Version No. 1.1

June 22, 16

Author: ICOE Team

**Document Control Information**

**Document Information**

|  |  |
| --- | --- |
| **Name** | Mulesoft Logging Standards |
| **Program Name** | Enterprise Integration |
| **Author** | Suchit Kavady, Rabi Sahu |
| **Version** | 1.2 |
| **Status** | Draft |

**Document Edit History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | Date | Description | Author |
| 1.0 | 02/17/2016 | Initial | Suchit Kavady |
| 1.1 | 04/21/2016 | updated | Rabi Sahu |
| 1.2 | 06/21/2016 | Updated the exceptions for CorrelationID in the use case table, in section 4. Added required logging patterns in a new section 5. | John Henschel |
| 1.3 | 6/22/2016 | Small changes to section 5 | John Henschel |

**Support / SME Resources**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team** | Name | Role | Contact (Email / Phone) |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of content

[1. Introduction 4](#_Toc443484542)

[2. High level System Interactions 4](#_Toc443484543)

[3. Logging Structure 5](#_Toc443484544)

[4. Recommended Logging Points 10](#_Toc443484545)

[5. Appendix 11](#_Toc443484546)

[a. Sample Log Entries 11](#_Toc443484547)

# Introduction

Mulesoft CloudHub is the integration platform to be used to meet Coca Cola’s Integration needs. Splunk will be used for monitoring and analytics of audit and exception logs generated by Mulesoft.

This document discusses the recommended standards for logging while building integration objects using Mulesoft. Though the document primarily focusses on Mulesoft CloudHub, the principles are applicable to integrations developed using On-Prem Mulesoft platform also.

# High level System Interactions

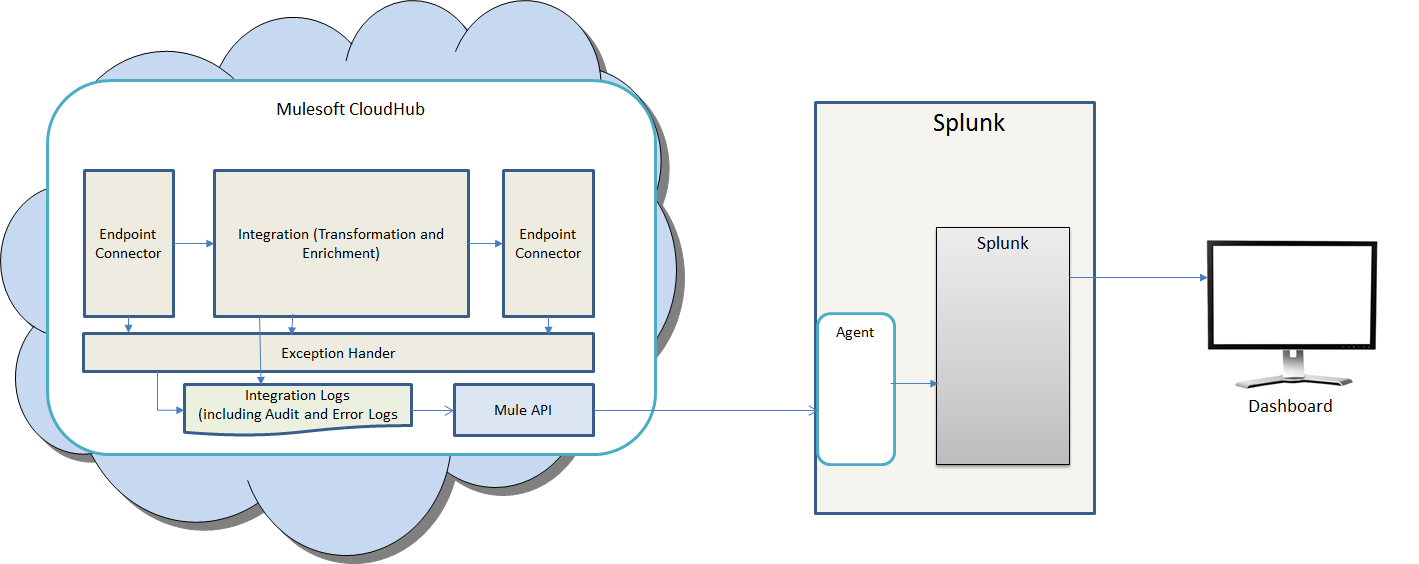
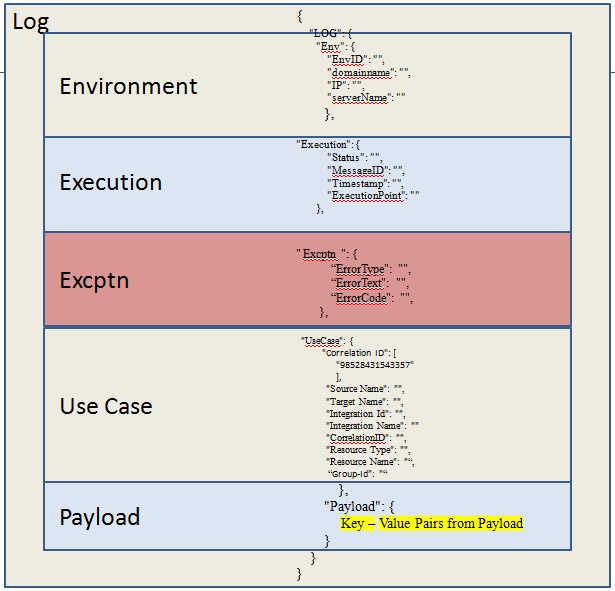


Figure above shows the high level system interactions between Mulesoft and Splunk.

Mulesoft CloudHub will write log entries to the log files. The log entries will include audit log entries as well as exception log entries.

An agent running on the Splunk server will use Mule API to download logs into Splunk for processing. Splunk will be configured to provide dashboards and drill down capabilities for users to view the data.

# Logging Structure



The data to be logged while writing log entries to Mule Log are divided into four sections.

1. Environment

Environment data is related to the environment which has processed the entry. All the entries for this section are mandatory. The entries include:

* EnvID
* Domainname
* IP
* serverName

1. Execution

Execution section captures the message metadata including:

* Status
* MessageID
* Timestamp
* ExecutionPoint

All entries in this section are mandatory

1. Excptn

Exception section is captured only when a technical or business exception has occurred. The entries include:

* ErrorType
* ErrorText
* ErrorCode
  1. Refer attached sheet for Codes



1. Use Case

Use case section captures data related to the message being processed.

The entries in this section include

|  |  |  |
| --- | --- | --- |
| Field | Description | Mandatory? |
| Source Name | Name of the source of the message. Ensure that the source name is consistent across all messages | Y (In a decoupled scenario the source could be ESB or S3 or SQS) |
| Target Name | Name of the Target of the message. Ensure that the Target name is consistent across all messages | Y (In a decoupled scenario the target could be ESB or S3 or SQS) |
| Integration ID | This should be the CI id of the integration object | Y |
| Integration Name | This should be the CI Name of the integration object | Y |
| Correlation ID | This should the unique id generated either by Apigee, within Mule Cloudhub or in Mule On-Prem.  Apigee generates 14 digit random number with ‘99’ as first 2 characters  In case Mule generates, it is of 14 digit random number with ‘98’ as first 2 characters  Exceptions:   1. While polling, when there is no message the correlationID will be ” “ 2. While doing message collection there could be multiple correlationIDs in an array. This should not be printed in the log entry, instead use “ “ and create an additional logging entry in the nearest FOREACH statement that will generate unique log entries for each correlationID. 3. There should never be an array of correlationIDs in a log entry. If this is the case, replace with “ “ and look for the next opportunity to provide a single log entry for each correlationID | Y |
| Resource Type | This should be the resource type that is currently being used when the log entry is created. See table below for the resource types (can be multiple comma separated entries) | Y |
| Resource Name | This should be the resource name that is currently being used when the log entry is created. See table below for the resource types (can be multiple comma separated entries) | Y |
| Group-Id | Applicable only in case of Mule sending file to end system. Populate the file name | N |

1. Payload

Payload section captures the payload being processed.

The following are the guidelines for the payload section

1. If the payload is available at the point at which the log entry is being captured, capture relevant key fields along with values from the payload. The key fields will depend on the particular business object being captured.

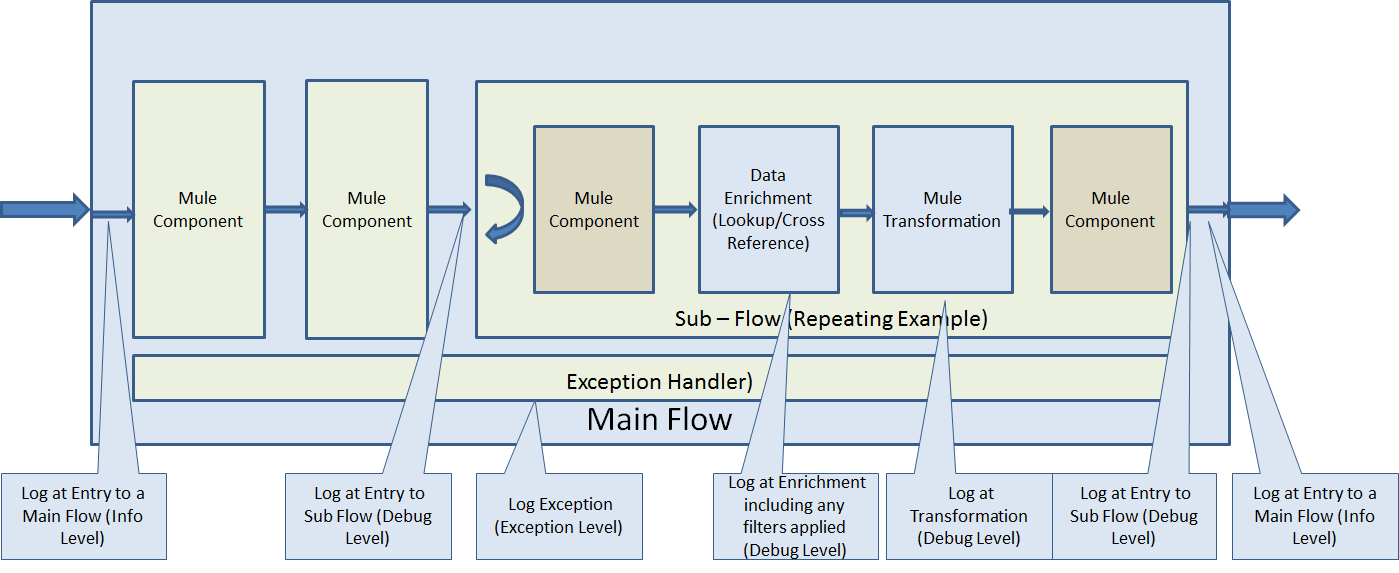
Typical key fields include business process or functional id, business object name (such as worker, purchase order, cost center etc.

1. If the payload is not available, capture process level details such as a queue name, file name, folder etc.

Table below shows the resource types and their names.

|  |  |  |
| --- | --- | --- |
| **Resource Type** | **Resource Name** | **Notes** |
| Data Translator | Data Map Source | Use this resource type if the log entry is being made during enrichment such as data lookup, cross referencing etc |
| Axway | Axway Destination File Name | Use this resource type if the log entry is being made during delivery to Axway |
| Transform | Transformation Object | Use this resource type if the log entry is being made during data transformation |
| S3 | S3 Filename (Bucket/Folder Name) | Use this resource while writing to or reading from S3 |
| SNS | SNS Topic Name | Use this resource while writing to SNS |
| SQS | SQS Queue Name | Use this resource while reading from SQS |
| Workday | Workday Connect, Workday Change | Use this entry when accessing data from Workday using connector on workday (not Mule workday connector). Workday Change is the connector used to send changes only. |
| Object Store | Object Store Reference Key | Use this resource while accessing Object store within Mule |
| SAP | SAP Host Name | Use this resource type if SAP is being accessed |
| IDOC | IDOC Type | This is a contingent entry and needs to be populated when SAP is being accessed |
| PGM ID | Program Id Value | This is a contingent entry and needs to be populated when SAP is being accessed |
| HTTP Listener | URL | Use this resource when accessing http end point as listener |
| Flow | Flow Name | Use this resource type when logging at the start of the flow and end of the flow |
| HTTP Call | SFDC URI | Use this resource type if SalesForce is being accessed |
| SFDC Connector | Org | This is a contingent entry and needs to be populated when SalesForce is being accessed |
| SFDC connector parameters – user id | user id value | This is a contingent entry and needs to be populated when SalesForce is being accessed |
| VM | VM path name | Entry for VM which is internal to Mule |

# Recommended Logging Points



Use the following guidelines for determining the logging points

1. Entry to a main flow has to be logged at a level of Info.
2. Exit from a main flow has to be logged at a level of info
3. Entry to a sub-flow has to be logged at a level of debug
4. Exit from a sub-flow has to be logged a level of debug
5. If the payload is not available when entering a flow, a log entry at the info level has to be created at the point when payload is available
6. If the sub-flow is iterative, log entries have to be made for each iteration
7. Log entry at debug level before and after a business object is modified (transformed)
8. Log entry at debug level when access external resources such as object store, lookup repository, S3, message systems etc.
9. Log exception with all data provided in the structure including exception section.

# Required logging patterns

* 1. Any time a mandatory/required field fails data validation, there should be a unique message in the ErrorText field of the log entry that specifies which field failed validation. The payload should contain a unique key value pair that can be used to identify the record in workday. The resource type should be “flow” and the resource name should be <flow name>. This pattern is an ERROR at the record level and will require that the ErrorType is set to “Field Validation Error”, ErrorText is set to “unique message that specifies which field(s) failed validation” and ErrorCode is set to “B0204”. Because this is a record level exception the rest of the file should be processed.
  2. Any time a record is dropped/skipped because a value doesn’t meet a filter criteria, there should be a log entry with the key value pairs for all fields used by the filter in the payload of the Splunk log entry. The exception to this is if the key value pair is for PI data. In this case provide the key and “PI” for the value. The payload should also contain a unique key value pair that can be used to identify the record in workday and this message; “Record dropped due to filter condition not matching.” If the record meets the filter criteria it should continue to the next component in the flow, if not it should go to an exception handler that generates the Splunk log entry. Because this is **not** a record level exception the exception block of the log entry should **not** contain values. A choice router will be used to generate an informational message if no file will be outputted because all records failed filter criteria.
  3. PI/SPI data that should be replaced with “PI” in the log entry’s
     1. Name
     2. E-mail address
     3. Phone number
     4. Bank account number
     5. Credit card number
     6. Passport number
     7. SSN

# Appendix

# Sample Log Entries

The following section shows details of the sample log entry at an integration point. The example shown below is captured at the beginning of transformation.

|  |
| --- |
| {  "LOG": {  "Env": {  "EnvID":"O65141-0921.. linke\_worker",  "domainname":"cocacola.com",  "IP":"192.168.5.156",  "serverName":"O65141-0921.na.ko.com"  },  "Execution":{  "Status":"Success",  "MessageID":"06a39eb0-c2df-11e5-b5a8-988120524153",  "Timestamp":" 2016-04-19 07:52:38:968 ",  "ExecutionPoint":" linke\_daily\_worker\_aggregated\_transformer\_flow starts”  },  "UseCase": {  "Correlation ID": [  "98528431543357"  ],  "Source Name": "Workday",  "Target Name": " LinkE ",  “Integration Id”: “UID000000338114”,  "Integration Name": " INT:HR-Worker-Daily-Full-Mobility-Sub "  "Resource Type": "Flow",  "Resource Name": " linke\_daily\_worker\_aggregated\_transformer\_flow "  },  "Payload": {  “Filter Criteria”: ““  “Employee\_Id” : 00001609”  “User\_Id” : “”  “HostCountry” : “USA”  “Terminated” : “False”  “Employment\_Term\_Date” : “”  “International\_Assignment” : “Yes”  “Worker\_Type” : “”  “Hire\_Date : “2015-09-23”  “Ledger\_Code” : “”  “Staffing\_Event” : “”  “National\_Id” : “”  “JobSeeker\_Id” : “”  “Primary\_Id” : “”  “Staffing\_Event” : “”  “Contingent\_Worker” : “”  “Sequence\_Number” : “1”  “Employee\_Personal\_Number” : “”  }  }  } |

The following sample captures the logging message in case of error during the transformation process:

|  |
| --- |
| {  "LOG": {  "Env": {  "EnvID": "O65141-0921.. linke\_worker ",  "domainname": "cocacola.com",  "IP": "192.168.5.156",  "serverName": "O65141-0921.na.ko.com"  },  "Execution": {  "Status": "Error",  "MessageID": "b902ac30-c2fe-11e5-8184-988120524153",  "Timestamp": " 2016-04-19 07:52:44.98",  "ExecutionPoint": "Error Handling"  },  "Excptn": {  "ErrorType": " A sequence of more than one item is not allowed as the first argument of normalize-space() (\"0.832\", \"Create Job Requisiti...\") (net.sf.saxon.trans.XPathException). Message payload is of type: String ",  "ErrorText": " A sequence of more than one item is not allowed as the first argument of normalize-space() (\"0.832\", \"Create Job Requisiti...\") (net.sf.saxon.trans.XPathException). Message payload is of type: String (org.mule.api.transformer.TransformerMessagingException). Message payload is of type: String ",  "ErrorCode": " T0904 "  },  UseCase": {  "Correlation ID": [  "98528431543357"  ],  "Source Name": "Workday",  "Target Name": " LinkE ",  “Integration Id”: “UID000000338114”,  "Integration Name": " INT:HR-Worker-Daily-Full-Mobility-Sub "  "Resource Type": "Flow",  "Resource Name": " linke\_daily\_worker\_transformation\_flow"  },  "Payload": {  “Filter Criteria”: ““  “Employee\_Id” : 00001609”  “User\_Id” : “”  “HostCountry” : “USA”  “Terminated” : “False”  “Employment\_Term\_Date” : “”  “International\_Assignment” : “Yes”  “Worker\_Type” : “”  “Hire\_Date : “2015-09-23”  “Ledger\_Code” : “”  “Staffing\_Event” : “”  “National\_Id” : “”  “JobSeeker\_Id” : “”  “Primary\_Id” : “”  “Staffing\_Event” : “”  “Contingent\_Worker” : “”  “Sequence\_Number” : “1”  “Employee\_Personal\_Number” : “””  }  }  } |