Oracle Financial Services Enterprise Case Management

API Services Guide

Release 8.1.2.0.0

June 2023

ORACLE
Financial Services

ORACLE

OFS ECM API Services Guide

Document Control

Version Number	Revision Date	Change Log
3.0	June 2023	Merged the API and Attachment & Comments Guides. See Attachment and Comments Service.
2.0	May 2022	ScenarioCatlogID and Its description changed to scenarioName.
1.0	April 2020	Created the first version.

Table of Contents

1 Preface

This preface provides supporting information for the Oracle Financial Services Enterprise Case Management Application Pack (OFS ECM) API Services.

Topics:

1.1 Purpose of this Document

This document contains the API services information for Oracle Financial Services Enterprise Case Management Application Pack. This API Services Guide aims to provide information about the REST APIs for ECM.

This guide assumes that the audience has technical and functional expertise in using and working with REST APIs. This document does not teach REST concepts.

1.2 Intended Audience

This document is intended for users of OFS ECM. You must be able to utilize the API service to populate events generated from an in-house application or any other third-party monitoring system.

1.3 Related Documents

This section identifies additional documents related to OFS ECM. You can access these documents from the Documentation Library (OHC).

- Oracle Financial Services Enterprise Case Management Installation Guide
- Oracle Financial Services Enterprise Case Management User Guide
- Oracle Financial Services Enterprise Case Management Admin Guide

2 Overview

OFS ECM API offers services to populate events identified in your source systems into the OFS ECM layer and services to allow the Behavior Detection Framework (BDF) to consume trusted pair information present within ECM.

In transaction monitoring systems, events are generated using various scenarios through batches. Then, the generated events are processed further, and cases are created. This whole process takes time. There may be some outliers for which a case can be created or extended to an existing case. This service helps users avoid the lengthier process of transaction monitoring systems. Also, these API services help Analysts create ad-hoc events if they come across any negative news on a customer or some other external incidents for an interested party for further investigation.

Trusted Pair is the concept of reducing the number of false positive events by identifying transactions between parties viewed as having a trusted relationship. The Trusted Pair API will allow full or filtered Trusted Pair data to be loaded to BDF based on inputs provided.

Event Suppression enables the automatic suppression of a particular entity's newly-generated alerts based on criteria such as highlight, scenario, and suppression rule begin and end date. The Event Suppression API will allow full or filtered Suppression data to be loaded to BDF based on inputs provided.

2.1 Process Flow

Figure 1 shows the process flow of how to use services.

Figure 1: Process Flow

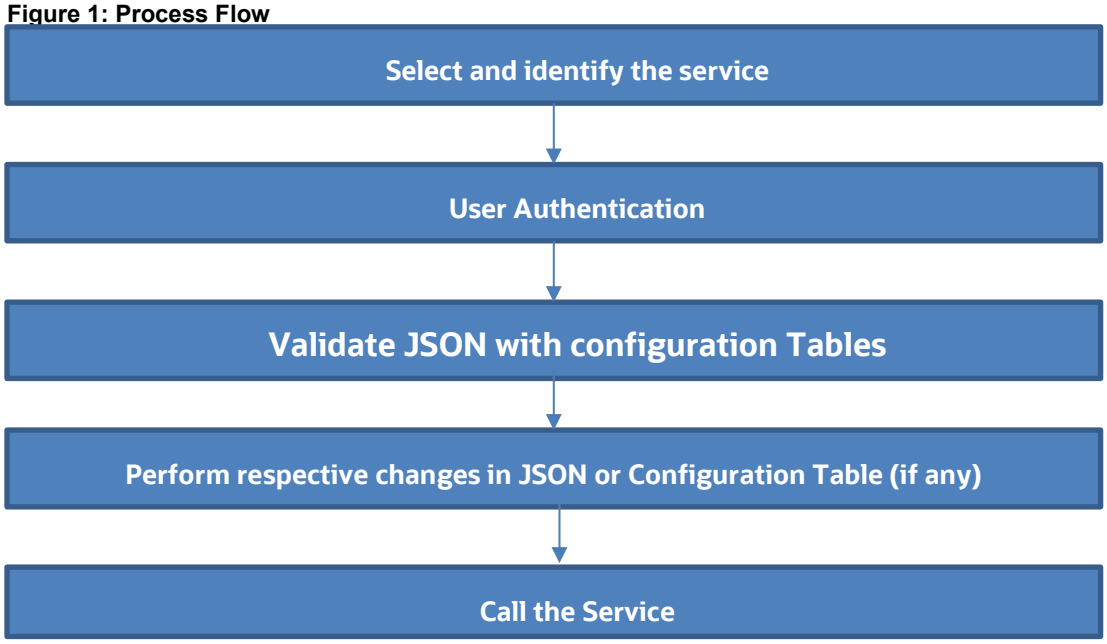


Table 1 provides additional information and links to specific documentation for each task in the flowchart.

Table 1: Task Details

Task	Details and Documentation
Select and identify the service	Select and identify the service which serves your purpose from the list of services. For more information, see Types of Services.
User Authentication	Identify the ECM user to use the service. User credentials must be set in the BASIC AUTH HEADER of the service request.
Validate JSON with configuration Tables	Compare the sample JSON with your required configuration. If it matches, then it can be used. Otherwise, it must be modified.
Perform respective changes in JSON or Configuration Table (if any)	Identify the necessary columns and add missing information to the configuration table as required. The columns added must be mapped back to the JSON. For more information, see Services.
Calling the Service	The Service is called using any REST Client.

2.2 Types of Services

The following services are supported:

- Create Event: This service saves the event generated in the source system to the consolidation layer. The saved event is available for correlation when the next batch is triggered.
- Create Event and Promote to Case: This service creates a new case for the event generated from the source system. For example, if an event is created for a customer in the Customer Screening application, on triggering this service, a case is created for the event on the ECM layer.
- Create Event and Extend to Existing Case: This service links a generated event to an existing case mentioned in the service. For example, an event generated in AML, which is related to another event for which a case is already created in ECM. On triggering this service, the new event is added to the existing case.
- Trusted Pairs - This service allows Behavior Detection Framework (BDF) to consume trusted pair information present within ECM. This API will have non-mandatory inputs. If these inputs are provided, they will be used to filter data. If these inputs are not provided, full data will be loaded as JSON output.
- Event Suppression - This service allows Oracle Financial Services Behavior Detection (OFS BD) to consume Event Suppression information present within ECM. This API will have non-mandatory inputs. If these inputs are provided, they will be used to filter data. If these inputs are not provided, full data will be loaded as JSON output.

3 Prerequisites

The following are prerequisites for using the API services:

1. ECM setup must be installed.
2. Appropriate User privileges to access the services.
3. Technical and functional knowledge to understand and execute the REST APIs and configuration knowledge.
4. Knowledge of REST concepts, JSON, and browser-based REST client.
5. Prior knowledge of REST programming is required to understand the examples, samples, scenarios, and reference sections.
6. Knowledge of the ECM Data model.

4 Configuring Tables

NOTE This section applies only to the following API Services

- Create Event
- Create Event and Promote to Case
- Create Event and Extend to Existing Case

You must configure specific tables to populate and save events on the Consolidation and Case layers. By configuring these tables, you will get the flexibility to select which data (such as Evented Entity and Entity details) should be saved. The Oracle client can configure these tables before utilizing the API service. The following tables must be configured:

- FCC_RT_EVENT_ENTITY_TABLES
- FCC_RT_EVENT_ENTITY_TABLES_COL
- FCC_RT_EVENTTYPE_PTC
- FCC_RT_EVENTTYPE_PTC_QUERIES
- FCC_RT_EVNTTYP_CORRELATION_MAP
- FCC_CORRELATION_CASE_TYPE_MAP

The following sections describe these tables.

4.1 FCC_RT_EVENT_ENTITY_TABLES

This table stores entities that the service accepts. For example, if this table has entries for only entities such as Account and Customer, then the API service will accept only events generated which have a focus on Account and Customer.

Column Name	Description
N_ENTITY_TYPE_ID	Entity Type unique ID.
V_ENTITY_TYPE	Entity type name. It should be the same as what is sent in the JSON. For example, Customer, Account, and so on.
V_EVENTED_TABLE_NAME	Name of the entity evented table where evented data is stored.
V_ENTITY_TABLE_NAME	Name of the corresponding CA business table name. For example, FCC_CUST.
V_ENTITY_LOOK_UP_TABLE_NAME	Name of the entity lookup table.
IS_LOOK_UP_PER_REQD	Determines if the entity details are required to copy to the entity lookup table. You can set this to Y or N. If flag is Y, then it will update the corresponding entity lookup tables. If you are adding a new entity and this flag is Y, then it will check the newly added entity in the lookup table and will not update it in the table if it is a duplicate.

Column Name	Description
IS_ENTI_PER_REQD	Determines if the entity details are required to copy to the entity table. You can set this to Y or N. If this flag is N, then entity information at the time of event generation will not be updated to the corresponding entity tables.
IS_EVENT_PER_REQD	Determines if the entity details are required to copy to the event table. That is, you can choose in which table data is to be populated. You can set this to Y or N. If this flag is N, then entity information at time of event generation will not be updated to the corresponding event tables.
V_LOOKUP_SEQUENCE_NAME	Sequence name to SKey generation of the lookup. This is required if lookup persistence is required. You can find the sequence from the Sequences (CM_CUST_SEQ) option on the left panel of the database.

4.2 FCC_RT_EVENT_ENTITY_TABLES_COL

This table maps the lookup, evented, and business entity information required to be captured for the respective entities, that is, the columns of the lookup, evented, and business entities that are required on the JSON. The mapping must match the data accepted in the relevant entity tables.

Column Name	Description
N_ENTITY_TYPE_ID	Sequence ID of the entity. It must be the same as defined in the FCC_RT_EVENT_ENTITY_TABLES table.
V_TABLE_NAME	Table name for the entity which is given in the FCC_RT_EVENT_ENTITY_TABLES table.
V_COL_BUSINESS_NAME	Business name of the entity attribute.
V_COL_NAME	Column name of the entity attribute.
V_DATA_TYPE	Data type of the entity attribute.
V_COL_TYPE	Column type of the entity attribute.
V_IS_SURROG_KEY	Defines the surrogate key.

4.3 FCC_RT_EVENTTYPE_PTC

This table maps the case type and corresponding T2T query IDs that must be triggered during case creation or case extension. These queries are triggered only if the service triggered is required to create or extend a case.

Column Name	Description
V_CASE_TYPE	The type of the case. For example, AML_DD
N_SEQUENCE	Order in which the queries should run for the corresponding case type.

Column Name	Description
V_T2T_CODE	Code name of the corresponding T2T query.

4.4 FCC_RT_EVENTTYPE_PTC_QUERIES

This table stores T2T queries against corresponding T2T codes. Queries are triggered on a case service call, but only the queries mapped to the case type for which the event is triggered will be called.

Column Name	Description
V_T2T_CODE	Code name of the corresponding T2T query.
C_T2T_QUERY	T2T query which populates data to the ECM tables.

4.5 FCC_RT_EVNTTYP_CORRELATION_MAP

This table maps the event type to the correlation rule, so the case generated through the service will be of case type (AML_SURV, CS_SAN, and so on) to which the correlation rule is mapped. For example, Correlation rule 1 is mapped to case type AML_SURV. If the ingesting AML event has to generate a case type of AML_SURV, the entry must be passed in this table mapping event type and correlation rule.

Column Name	Description
N_EVENT Type	The type of event used for correlation.
N_CORRELATION_RULE_SKEY	This is the correlation rule unique Identification number. The value of N_CORRELATION_RULE_SKEY column (rule number) should be the same as defined in the FCC_CORRELATION_RULE table.

4.6 FCC_CORRELATION_CASE_TYPE_MAP

This table maps the correlation rule with the case type, so cases created through this correlation rule will be all of the given case type (for more information on how to design the case type, see the Oracle Financial Services Enterprise Case Management Admin Guide).

Column Name	Description
V_CASE_TYPE	This is the type of case.
N_CORRELATION_RULE_SKEY	This is the correlation rule unique Identification number. The value of N_CORRELATION_RULE_SKEY column (rule number) should be same as defined in the FCC_CORRELATION_RULE table.

5 Use Cases for Configuring Tables

NOTE This section applies only to the following API Services

- Create Event
- Create Event and Promote to Case
- Create Event and Extend to Existing Case

You can use the following scenarios to understand the services. Before modifying the JSON, you must configure the tables with appropriate values.

Adding an Entity and Adding Persistence to an Entity use cases explain how to add a new entity in FCC_RT_EVENT_ENTITY_TABLES.

Adding an Entity, Adding Persistence to an Entity, and Adding PTC for Entity use cases explain how to use the Create Event service.

Adding an Entity, Adding Persistence to an Entity, and Adding PTC for Entity, and Adding an Event Type to Correlation Map use cases explain how to create Event and Attach to Existing Case.

Adding an Event Type to Correlation Map use case explains how to map an event type to correlation rule.

This section includes the following topics:

5.1 Adding an Entity

To add an entity, follow these steps:

1. Add a new entry in the FCC_RT_EVENT_ENTITY_TABLES table, as shown in the following example:

N_ENTITY_TYPE_ID	V_ENTITY_TYPE	V_EVENTED_TABLE_NAME	V_ENTITY_TABLE_NAME	V_ENTITY_LOOK_UP_TABLE_NAME	IS_LOOK_UP_PER_REQD	IS_ENTI_PER_REQD	IS_EVENT_PER_REQD	V_LOOKUP_SEQUENCE_NAME
102	ECM_CUSTOMER	FCC_CUST_EVNT	FCC_CUST	FCC_CUSTOMER_LOOKUP	Y	Y	Y	

Oracle recommends setting the flag to Y for IS_LOOK_UP_PER_REQD and IS_EVENT_PER_REQD columns. V_ENTITY_TABLE_NAME table names must be the same as mentioned in the backend ECM data model.

2. Add the evented details in the FCC_RT_EVENT_ENTITY_TABLES_COL table for the newly added entity, as shown in the following example:

N_ENTITY_TYPE_ID	V_TABLE_NAME	V_COL_BUSINESS_NAME	V_COL_NAME	V_DATA_TYPE	V_COL_TYPE	V_IS_SURROG_KEY
102	FCC_CUST_EVNT	custId	CUST_INTRL_ID	CHAR		

Here, the entry in the V_TABLE_NAME column must be the same as either the V_EVENTED_TABLE_NAME, V_ENTITY_TABLE_NAME, or V_ENTITY_LOOK_UP_TABLE_NAME column of the FCC_RT_EVENT_ENTITY_TABLES table.

5.2 Adding Persistence to an Entity

For the FCC_RT_EVENT_ENTITY_TABLES table, the following scenarios are possible:

- If the IS_LOOK_UP_PER_REQD, IS_ENTI_PER_REQD, and IS_EVENT_PER_REQD columns are set to Y, then the corresponding Entity and Event tables will be populated after lookup.

N_ENTITY_TYPE_ID	V_ENTITY_TYPE	V_EVENTED_TABLE_NAME	V_ENTITY_TABLE_NAME	V_ENTITY_LOOK_UP_TABLE_NAME	IS_LOOK_UP_PER_REQD	IS_ENTI_PER_REQD	IS_EVENT_PER_REQD	V_LOOKUP_SEQUENCE_NAME
102	ECM_CUSTOMER	FCC_CUST_EVNT	FCC_CUST	FCC_CUSTOMER_LOOKUP	Y	Y	Y	

- If the IS_LOOK_UP_PER_REQD and IS_EVENT_PER_REQD columns are set to Y, then the corresponding Event tables will be populated after lookup. If the IS_ENTI_PER_REQD column is set to N, then the UI may not display the correct data.

N_ENTITY_TYPE_ID	V_ENTITY_TYPE	V_EVENTED_TABLE_NAME	V_ENTITY_TABLE_NAME	V_ENTITY_LOOK_UP_TABLE_NAME	IS_LOOK_UP_PER_REQD	IS_ENTI_PER_REQD	IS_EVENT_PER_REQD	V_LOOKUP_SEQUENCE_NAME
102	ECM_CUSTOMER	FCC_CUST_EVNT	FCC_CUST	FCC_CUSTOMER_LOOKUP	Y	N	Y	

5.3 Adding PTC for Entity

To add PTC for an entity, follow this step:

Add a new entry in the FCC_RT_EVENTTYPE_PTC table. Before adding the entries in the FCC_RT_EVENTTYPE_PTC table, populate the FCC_RT_EVENT_ENTITY_TABLES and FCC_RT_EVENT_ENTITY_TABLES_COL tables.

When a case type is added, the corresponding KDD_CASE tables will be populated.

The following example shows how to add new entries:

V_CASE_TYPE	N_SEQUENCE	V_T2T_CODE
AML_DD	1	t2t_KDD_CASE_ACCOUNTS
AML_DD	2	t2t_KDD_CASE_CUSTOMERS
AML_DD	3	t2t_KDD_CASE_INSTN_MASTER
AML_DD	4	t2t_KDD_CASE_CLIENT_BANK
AML_DD	5	t2t_KDD_CASE_EXTERNAL_ENTITY

5.4 Adding an Event Type to Correlation Map

To add an event type to the correlation map, follow this step:

Add a new entry in the FCC_RT_EVNTTYP_CORRELATION_MAP table, as shown in the following example:

N_CORRELATION_RULE_SKEY	V_EVENT_TYPE
12	AML_SURV

6 Services

The following services are available:

- Create Event
- Create Event and Promote to Case
- Create Event and Extend to Existing Case
- Get Trusted Pair Information
- Get Event Suppression Information
- Create or update Case description
- Get ECM Case Status
- Add a customer to a case
- Remove Account from case
- Add Account to a case
- Add external entity to a case

6.1 Create Event

This service saves the event generated in the source system to the consolidation layer. The saved event will be available for correlation when the next batch is triggered.

The following topics explain how to configure and use the Create Events in JSON.

- HTTP Link
- Service Type
- Request Parameters
- Request JSON Sample
- Response Parameters
- Response JSON Sample

6.1.1 HTTP Link

http:// <Application URL>/rest-api/CMRestService/RealTimeCaseCreationService/saveEvents

6.1.2 Service Type

The service type is POST.

6.1.3 Request Parameters

The following table describes the details of the parameters.

First Level	Second Level	Third Level	Details
events			This accepts basic information of the Event.
events	eventCode		This accepts event ID information.
events	eventScore		This accepts information about the event score.
events	dataOrigin		This accepts information about the Data Origin of the event. It stores the source system from where data is extracted.
events	jurisdictionCode		This accepts information about the Jurisdiction code of the event. It represents geographic or legal entity monitoring or data restrictions.
events	businessDomain		This accepts information about the Business Domain of the event. Business Domain represents the operational data restrictions (that is, potential Chinese walls between business units).
events	type		This accepts information about the 'Type' of the event. For example, AML, KYC, and so on.
events	priority		This accepts information about the 'Priority' of the event. It is the Priority of the Event and the risk it poses to a customer or an institution. low, medium, high
events	scenarioClass		This accepts information about the 'Scenario Class' of the event. For example, ML, FR, and so on.

First Level	Second Level	Third Level	Details
events	comments		This accepts the additional comments provided, which are related to the generated Event, and details of the event or assignment that led to the creation of an external event that is formatted for display.
events	entities		This is a second-level tag under JSON. It accepts basic information-related event and entity mapping.
events	entities	entityCode	This accepts the entity code of the entity. It is the Entity identifier on whom the event is generated from the external system. It can be alphanumeric. For Example, Customer identifier, Account identifier on whom the event is generated.
events	entities	entityType	This accepts the entity type of the entity. For example, Account, Customer, Employee, External_Entity, and Household.
events	entities	entityTypeCode	This accepts the entity type code of the entity.
events	entities	entityName	This accepts the name of the entity.
events	entities	focusFlag	This accepts whether the entity is the Primary Focus of the event or not. List of Values are Y or N. Only one entity of an event will have the value of Y. All others will have N.
events	entities	actId	This accepts the account internal ID of the Account.

First Level	Second Level	Third Level	Details
events	entities	genericEntityInformation	This accepts the generic entity information in the form of a JSON value. Note that a generic entity can never be a focal entity. Hence, there must be some other entity also specified in the entity array, which is the focal entity for the event. For more samples, see Appendix A.
events	details		This is a second-level tag under JSON. It accepts additional details related to the event.
events	details	reason	This accepts event highlights.
events	details	scenarioName	This accepts the event scenarioName.
events	bindings	bindingName	This accepts the event binding name.
events	bindings	bindingActualValue	This accepts the actual binding value.
events	bindings	bindingBaseValue	This accepts the binding base value.
events	bindings	bindingType	This accepts the binding type.

6.1.4 Request JSON Sample

This section contains a request JSON sample for creating events.

The entries in this sample are only for reference purposes.

```
{
                "events": [{
                                "eventCode": "09875456",
                                "eventScore": "",
                                "dataOrigin": "DLY",
                                "jurisdictionCode": "AMEA",
                                "businessDomain": "c",
                                "type": "AML_SURV",
                                "priority": "",
                                "scenarioClass": "AML",
```

```
                                "comments": "",
                                "entities": [{
                                                "entityCode": "XXXACFTNEXTENAC-
123007",
                                                "entityType": "ACCOUNT",
                                                "entityTypeCode":
"ECM_ACCOUNT",
                                                "entityName": "BARRY",
                                                "focusFlag": "Y",
                                                "actId": "XXXACFTNEXTENAC-
123007"
                                },
{
            "genericEntityInformation": {
                                "Customer Details": {
                                "Victim Name": "ABC XYZ",
                                "Victim SSN": "123-12-1234",
                                "Phone Number": "123-123-1234"
                                            },
                                "Enterprise Customer Complaint": {
                                "Was a complaint filed": "No"
                                            },
                                "Incident Details": {
                               "Incident Type": "Fradulent Transaction",
                                "Account Type": "Consumer",
                                "Market": "Southwest Mumbai"
                                            }
                    }
        }],

                                "details": [{
                                          "reason": "INR2000        Suspicious
Transaction to Terror Outfits",
                                                "scenarioName": "User Defined
Event"
                                }],
                                "bindings": [{
                                                "bindingName":
"Effectv_Risk_Lvl",
```

```
                                        "bindingActualValue": "50",
                                        "bindingBaseValue": "5",
                                        "bindingType": ""
                                }]
                        }],
                        "entities": [{
                                "entityCode": "XXXACFTNEXTENAC-123007",
                                "entityType": "ACCOUNT",

"jurisdiction": "AMEA",
                                "entityTypeCode": "ECM_ACCOUNT",
                                "entityName": "BARRY",
                                "actId": "XXXACFTNEXTENAC-123007",
                                "dataOrigin": "DLY",
                                "accName": "BARRY",
                                "types": "SAV",
                                "registration": "IND",
                                "ownerType": "",
                                "status": "A",
                                "branch": "BRANCH-001",
                                "taxCode": "N",
                                "householdGroupId": "",
                                "pimeBroker": "N",
                                "taxId": "T",
                                "firmId": "MANTAS",
                                "legalEntity": "BGMC_ACCT_001",
                                "statusEffective": "31-12-2014 00:00:00",
                                "lastActivity": "30-11-2015 00:00:00",
                                "openData": "31-12-2014 00:00:00",
                                "closeDate": ""
                        }]
}
```


NOTE From Oracle Financial Services Enterprise Case Management Release 8.1.2.2.0, capture of Generic information about entities such as Customer, Account and External entity are supported as part of Event Creation. Refer Appendix C for Sample JSONs.

6.1.5 Response Parameters

The following table describes the details of the response parameters.

First Level	Details
Message	Displays the message of service execution.
Status	Displays the status as Success or Fail.
Event Id(s)	Displays all Event IDs that were passed in the request JSON. Will show multiple event IDs if more than one was passed in input JSON.

6.1.6 Response JSON Sample

This section contains a response JSON sample for creating events.

The entries in this sample are only for reference purposes.

```
{
    "MESSAGE": "Events are saved successfully.",
    "STATUS": "SUCCESS",
"Event Id(s)": "09875456"
}
```

6.2 Create Event and Promote to Case

This service creates a new case for the event generated from the source system. For example, if an event is created for a customer in the Customer Screening application, on triggering this service, a case is created for the event on the ECM layer.

The defaulted priority for the cases created through the Real Time Case Creation service will be Low. The system will not determine the priority of the cases created through the service based on the Case Priority configuration.

Due date is updated when a case is created using a real-time API.

The following topics explain how to configure and use the Create Event and Promote to Case service in JSON:

- HTTP Link
- Service Type
- Request Parameters
- Request JSON Sample
- Response Parameters
- Response JSON Sample

6.2.1 HTTP Link

http:// <Application URL>/restapi/CMRestService/RealTimeCaseCreationService/saveEventsAndPromoteToCase

6.2.2 Service Type

The service type is POST.

6.2.3 Request Parameters

The following table describes the details of the parameters.

First Level	Second Level	Third Level	Details
events			This accepts basic information about the Event.
events	eventCode		This accepts event ID information.
events	eventScore		This accepts information about the event score.
events	dataOrigin		This accepts information about the Data Origin of the event. It stores the source system from where data is extracted.
events	jurisdictionCode		This accepts information about the Jurisdiction code of the event. It represents geographic or legal entity monitoring or data restrictions.
events	businessDomain		This accepts information about the Business Domain of the event. Business Domain represents the Customer's operational data restrictions (that is, potential Chinese walls between business units).
events	type		This accepts information about the 'Type' of the event. For example, AML, KYC, and so on.

First Level	Second Level	Third Level	Details
events	priority		This accepts information about the 'Priority' of the event. It is the Priority of the Event and the risk it poses to a customer or an institution. List of values: Low Medium High
events	scenarioClass		This accepts information about the 'Scenario Class' of the event. For example, ML, FR, and so on.
events	comments		This accepts the additional comments provided, which are related to the generated Event, and details of the event or assignment that led to the creation of an external event that is formatted for display.
events	entities		This is a second-level tag under JSON. It accepts basic information-related event and entity mapping.
events	entities	entityCode	This accepts the entity code of the entity. It is the Entity identifier on whom the event is generated from the external system. It can be alphanumeric. For Example, The Customer identifier or Account identifier on whom the event is generated.
events	entities	entityType	This accepts the entity type of the entity. For Example, Account, Customer, Employee, External_ Entity, and Household.
events	entities	entityTypeCode	This accepts the entity type code of the entity.
events	entities	entityName	This accepts the name of the entity.

First Level	Second Level	Third Level	Details
events	entities	focusFlag	This accepts whether the entity is the Primary Focus of the event or not. List of Values are Y or N. Only one entity of an event will have the value of Y. All others will have N.
events	entities	actId	This accepts the account internal ID of the Account.
events	entities	genericEntityInformation	This accepts the generic entity information in the form of a JSON value. Note that a generic entity can never be a focal entity. Hence, there must be some other entity also specified in the entities array which is the focal entity for the event. For more samples, see Appendix A.
events	details		This is a second-level tag under JSON. It accepts additional details related to the event.
events	details	reason	This accepts event highlights.
events	details	scenarioName	This accepts the event scenarioName.
events	bindings	bindingName	This accepts the event binding name.
events	bindings	bindingActualValue	This accepts the actual binding value.
events	bindings	bindingBaseValue	This accepts the binding base value.
events	bindings	bindingType	This accepts the binding type.
casePriority			This accepts the priority of the case.

6.2.4 Request JSON Sample

This section contains a Request JSON sample for creating an event and promote to case.

The entries in this sample are only for reference purposes.

```
{
                "events": [{
                                "eventCode": "655867843",
                                "eventScore": "",
```

```
                                "dataOrigin": "DLY",
                                "jurisdictionCode": "AMEA",
                                "businessDomain": "a",
                                "type": "AML_SURV",
                                "priority": "",
                                "scenarioClass": "AML",
                                "comments": "",
                                "entities": [{
                                                "entityCode": "1412",
                                                "entityType": "CLIENT_BANK",
                                                "entityTypeCode":
"ECM_CLIENT_BANK",
                                                "entityName": "JADEJ",
                                                "focusFlag": "Y",
                                                "actId": "XXXACFTNEXTENAC-
123007"
                                }
{
              "genericEntityInformation": {
                                "Customer Details": {
                                "Victim Name": "ABC XYZ",
                                "Victim SSN": "123-12-1234",
                                "Phone Number": "123-123-1234"
                                                },
                                "Enterprise Customer
                                Complaint": {
                                "Was a complaint filed?": "No"
                                                },
                                "Incident Details": {
                             "Incident Type": "Fradulent Transaction",
                                          "Account Type": "Consumer",
                                           "Market": "Southwest Mumbai"
                                                }
                                }
                           }],

                                "details": [{
"reason": "INR7890 TEST - CORRESPONDENT BANK",
```

```
                                        "scenarioName": "User Defined
Event"
                                }]
                        }],
                        "entities": [{
                                "entityCode": "1412",
                                "entityType": "CLIENT_BANK",
                                "entityTypeCode": "ECM_CLIENT_BANK",
                                "entityName": "JADEJ",
                                "instId": "HSINDI-001",
                                "instTypeCd": "BIC",
                                "dataOrigin": "DLY",
                                "city": "",
                                "country": "",
                                "postalCode": "",
                                "state": "",
                                "address": "",
                                "instituteCountry": "",
                                "instituteName": "JADEJ",
                                "largeInstitution": "N",
                                "selfInstitution": "N"
                        }, {
                                "skey": "1412",
                                "entityType": "CLIENT_BANK",
                                "entityTypeCode": "ECM_CLIENT_BANK_SEC",
                                "dataOriginCb": "DLY",
                                "cbEfctvRiskNb": "0",
                                "cbListSrcCd": "",
                                "cbMatchTx": "",
                                "busUnitCd": "",
                                "acctMgrNm": "",
                                "lastActDate": "31-12-2014 00:00:00",
                                "cbListRiskNb": "0",
                                "cbMatchTypecd": "",
                                "instSeqId": "102",
                                "jurisdiction": "AMEA",
                                "busList": "a"
                        }]
```

```
"casePriority":"High"
}
```

> **NOTE** From Oracle Financial Services Enterprise Case Management Release 8.1.2.2.0, capture of Generic information about entities such as Customer, Account and External entity are supported as part of Event Creation. Refer Appendix C for Sample JSONs.

6.2.5 Response Parameters

The following table describes the details of response parameters.

First Level	Details
CaseId	Displays the case ID.
Message	Displays the message of service execution.
Status	Displays the status as Success or Fail.
Event Id(s)	Displays all Event IDs that were passed in the request JSON. Will show multiple event IDs if more than one was passed in input JSON.
CaseType	Displays the type of case.

6.2.6 Response JSON Sample

This section contains a Response JSON sample for creating an event and promoting to case.

The entries in this sample are only for reference purposes.

```
{
    "caseId": "CA354",
    "message": "Events are saved successfully and promoted to case ",
    "status": "SUCCESS",
    "Event Id(s)": "655867843",
    "caseType": "AML_SURV"
}
```

6.3 Create Event and Extend to Existing Case

This service links a generated event to an existing case mentioned in the service. For example, an event generated in AML that is related to another event for which a case is already created in ECM. On triggering this service, the new event will be added to the existing case.

The following topics explain how to configure and use the Create Event and attach it to an existing Case service in JSON:

6.3.1 HTTP Link

http:// <Application URL>restapi/CMRestService/RealTimeCaseCreationService/saveEventsAndExtendToCase

6.3.2 Service Type

The service type is POST.

6.3.3 Request Parameters

The following table describes the details of the parameters.

First Level	Second Level	Third Level	Details
events			This accepts basic information about the Event.
events	eventCode		This accepts event ID information.
events	eventScore		This accepts information about the event score.
events	dataOrigin		This accepts information about the Data Origin of the event. It stores the source system from where data is extracted.
events	jurisdictionCode		This accepts information about the Jurisdiction code of the event. It represents geographic or legal entity monitoring or data restrictions.
events	businessDomain		This accepts information about the Business Domain of the event. Business Domain represents the operational data restrictions (that is, potential Chinese walls between business units).

First Level	Second Level	Third Level	Details
events	type		This accepts information about the 'Type' of the event. For example, AML, KYC, and so on.
events	priority		This accepts information about the 'Priority' of the event. It is the Priority of the Event and the risk it poses to a customer or an institution. List of values: Low Medium High
events	scenarioClass		This accepts information about the 'Scenario Class' of the event. For example, ML, FR, and so on.
events	comments		This accepts the additional comments provided, which are related to the generated Event, and details of the event or assignment which led to the creation of an external event that is formatted for display.
events	entities		This is a second-level tag under JSON. It accepts basic information-related event and entity mapping.
events	entities	entityCode	This accepts the entity code of the entity. It is the Entity identifier on whom the event is generated from the external system. It can be alphanumeric. For Example, The Customer identifier or Account identifier on whom the event is generated.
events	entities	entityType	This accepts the entity type of the entity. For Example, Account, Customer, Employee, External_ Entity, and Household
events	entities	entityTypeCode	This accepts the entity type code of the entity.

First Level	Second Level	Third Level	Details
events	entities	entityName	This accepts the name of the entity.
events	entities	focusFlag	This accepts whether the entity is the Primary Focus of the event or not. List of Values are Y or N. Only one entity of an event will have the value of Y. All others will have N.
events	entities	actId	This accepts the account internal ID of the Account.
events	entities	genericEntityInformation	This accepts the generic entity information in the form of a JSON value. Note that a generic entity can never be a focal entity. Hence, there must be some other entity also specified in the entities array, which is the focal entity for the event. For more samples, see Appendix A.
events	details		This is a second-level tag under JSON. It accepts additional details related to the event.
events	details	reason	This accepts the event highlights.
events	details	scenarioName	This accepts the event scenarioName.
events	bindings	bindingName	This accepts the event binding name.
events	bindings	bindingActualValue	This accepts the actual binding value.
events	bindings	bindingBaseValue	This accepts the binding base value.
events	bindings	bindingType	This accepts the binding type.

6.3.4 Request JSON Sample

This section contains a request JSON sample for creating an event and attaching to case:

The entries in this sample are only for reference purposes.

```
{
  "events": [{
        "eventCode": "999979",
        "eventScore": "",
        "dataOrigin": "DLY",
```

```
        "jurisdictionCode": "AMEA",
        "businessDomain": "c",
        "type": "AML",
        "priority": "",
        "scenarioClass": "AML",
        "comments": "",
  "entities": [{
"entityCode": "ACDPGTOTACTAC-
010",
"entityType": "ACCOUNT",
"entityTypeCode":
"ECM_ACCOUNT",
"entityName": "ELEVENTH",
        "focusFlag": "Y",
"actId": "ACDPGTOTACTAC-
010"
  {
             "genericEntityInformation": {
                   "Bank Details": [{
                         "Name": "HHHH",
                         "Year": 1988,
                         "Founder":"ABC XYZ",
                   "Departments": [{
                         "Name": "Credit",
                         "Start Year": "1989",
                         "Employees": [{
                               "Name": "XXX",
                               "City": "Bangalore"
                               }, {
                               "Name": "XXX",
                               "City": "Jaipur"
                               }]
                               },
                               {
                               "Name": "Debit",
                               "Start Year": "1989",
                               "Employees": [{
                               "Name": "XXX",
```

```
                              "City": "Bangalore"
                              }, {
                                    "Name": "XXX",
                                    "City": "Kandy"
                              }]
                        }
                  ]
            },
            {
                  "Name": "IIII",
                  "Year": 1992,
                  "Founder": "ABC XYZ",
                  "Departments": [{
                              "Name": "Credit",
                              "Start Year": "1993",
                              "Employees": [{
                                    "Name": "XXX",
                                    "City": "Mumbai"
                              }, {
                                    "Name": "XXX",
                                    "City": "Delhi"
                              }]
                        },
                        {
                              "Name": "Debit",
                              "Start Year": "1993",
                              "Employees": [{
                                    "Name": "XXX",
                                    "City": "Sydney"
                              }, {
                                    "Name": "XXX",
                                    "City": "Colombo"
                              }]
                        }
                  ]
            }
      ]
```

```
        }


  }],
  "details": [{
"reason": "INR2000 Suspicious Transaction to Terror
Outfits",
"scenarioName": "User Defined Event"
  }],
  "bindings": [{
"bindingName":
"Effectv_Risk_Lvl",
"bindingActualValue":
"50",
"bindingBaseValue": "5",
        "bindingType": ""
  }]
  }],
  "entities": [{
        "entityCode": "ACDPGTOTACTAC-
010",
        "entityType": "ACCOUNT",
        "entityTypeCode": "ECM_ACCOUNT",
        "entityName": "ELEVENTH",
        "actId": "ACDPGTOTACTAC-010",
        "dataOrigin": "DLY",
        "accName": "ELEVENTH",
        "types": "INV",
        "registration": "IND",
        "ownerType": "RPT",
        "jurisdiction": "AMEA",
        "status": "A",
        "branch": "BRANCH-001",
        "taxCode": "N",
        "householdGroupId": "HHSAMPLE00",
        "pimeBroker": "N",
        "taxId": "S",
        "firmId": "",
        "legalEntity": "BGMC_ACCT_001",
```

```
"statusEffective": "31-12-2014
00:00:00",
        "lastActivity": "",
        "openData": "31-12-2014
00:00:00",
        "closeDate": ""
              "events":[
      {
         "eventCode":"153803908863599",
         "eventScore":"",
         "dataOrigin":"DLY",
         "jurisdictionCode":"AMEA",
         "businessDomain":"c",
         "type":"AML",
         "priority":"",
         "scenarioClass":"AML",
         "comments":"",
         "entities":[
            {
               "entityCode":"CUHRTHRCPHH-002",
               "entityType":"CUSTOMER",
               "entityTypeCode":"ECM_CUSTOMER",
               "entityName":"ANDREW ELEVENTH",
                "jurisdiction" :"AMEA",
               "focusFlag":"Y",
               "custId":"CUHRTHRCPHH-002"
            }
         ],
         "details":[
            {
               "reason":"ffjfjgghgkkkkhhhk",
               "scenarioName":"User Defined Event"
            }
         ]
      }
   ],
   "entities":[
      {
```

```
            "entityCode":"CUHRTHRCPHH-002",
            "entityType":"CUSTOMER",
            "entityTypeCode":"ECM_CUSTOMER",
            "entityName":"ANDREW ELEVENTH",
            "jurisdiction" :"AMEA",
            "custId":"CUHRTHRCPHH-002",
            "dataOrigin":"DLY",
            "customerName":"ANDREW ELEVENTH",
            "customerType":"IND",
            "taxID":"TAXHRTHRCPHH-002",
            "legalStructCode":"GOV",
            "aliasName":"",
            "incomeRange":"",
            "industry":"US",
            "publicCompany":"",
            "estNetWorth":"1182552",
            "atmDailyLimit":null,
            "estAnualIncome":"168936",
            "added":"31-12-2014 00:00:00",
            "incorporationDate":"",
            "customerstatus": "A",
            "effectiverisk": "2",
            "busdomain": "a"
        }
    ],
            "caseID": "CA115"
}
```


> **NOTE** From Oracle Financial Services Enterprise Case Management Release 8.1.2.2.0, capture of Generic information about entities such as Customer, Account and External entity are supported as part of Event Creation. Refer Appendix C for Sample JSONs.

6.3.5 Response Parameters

The following table describes the details of the response parameters.

First Level	Details
Message	Displays the message of service execution.
Status	Displays the status as Success or Fail.
Event Id(s)	Displays all Event IDs that were passed in the request JSON. Will show multiple event IDs if more than one was passed in input JSON.
CaseType	Displays the type of case.

6.3.6 Response JSON Sample

This section contains a response JSON sample for creating an event and attaching to case.

The entries in this sample are only for reference purposes.

```
{
    "message": "Events are saved successfully and promoted to case ",
    "status": "SUCCESS",
    "Event Id(s)": "999979",
    "caseType": "AML_SURV"
}
```

6.4 Attachment and Comments Service

This ECM service allows clients to post attachments and comments from an external application to a case in Oracle's Enterprise Case Management application. This is useful when there are external processes that create evidence for a case and that information needs to automatically be attached to the case. For example, a client may have an RPA process which creates a customer profile for an investigator which will assist in the investigation. This service can post that profile to the case. It is then viewable from the Evidence tab within the case. The process is also recorded in the audit history of the case. The receipt of the attachment or comment may also be a status changing action. This allows for the status of the case to be updated when the document/comment is received.

Note that this service is only available in ECM release 8.0.7 and beyond.

With this service, you can associate attachments or comments with a case, a business entity, or both. At this time, attachments can only be viewed through a case. The ability to view a business entity and see attachments and comments related to that entity will be provided in a future release. If you intend these attachments and comments to be for an entity, Oracle recommends you associate to both cases and entities in preparation for this future functionality.

Each request can only associate a document or comment with one case or one entity.

Topics:

6.4.1 Service Definition

Topics:

6.4.1.1 Service URL

In the following example, a customer profile document created on customer John Doe (Customer ID CA12345) is being attached to case CA54321. The source of the document is from an RPA application. When the attachment is received, the status of the case will update to the status associated with action code CA986 as defined in PMF.

```
http://<<webserver IP or name>>/<<application context>>/rest-
api/ECMService/CaseManagementService/ecmUploadFile?entityId=CA54321&caseId=
CA54321&entityType=case&comments=Customer Profile for John Doe&docDesc=Customer
Profile from RPA Process&docType=Customer Profile&source=RPA Process ID
1234&senderId=RPAUser1234&actionCode=CA986
```

NOTE Here, **&** is used for separating the parameters in the above URL. The document or file has to be attached as multipart/form-data while calling the service. Authentication information has to be provided with Bas64 encoded format of <username>:<password>. Any space or special characters in the URL query parameters need to be 'Percent-encoded'

6.4.1.2 Service Method

The service method is POST for the enhanced security.

6.4.1.3 Request/Query Parameters

The following table describes the details of parameters.

Name	Parameter Name	Description	Required
Entity ID	entityId	It can be the Case ID or the Business Entity ID to which the document/comment is associated. If entity type is 'CASE', then entityId will be validated against the valid case IDs in the system. No validation for other entity types.	Y

Name	Parameter Name	Description	Required
Entity Type	entityType	Entity type is the type of entity against which this action is performed. By default, entity type 'CASE' is defined in the system. User needs to configure the other entity type in the system. For configuring new entity type, refer section Configuring Entity Type. If entity type is other than CASE, then provide an entity ID in the entityId parameter. You can also provide a case ID in the caseId parameter, so the document/comment will be associated with both case and the entity.	Y
Comments	comments	Comments to be stored against the Case/Entity. You can enter up to 64k characters.	Y
Sender ID	senderId	This can be a valid ECM user ID or a valid AAI user ID. If it is a valid ECM user ID, this will be displayed in the audit history and Evidence tab. If it is a valid AAI ID, then 'SYSTEM' will be displayed in the Audit History and Evidence tabs. If using an ECM user ID it will go against the client's ECM licensed user count. If using an AAI ID, it is recommended that some user identifier is also included in either the Document Description or Comment parameters.	N
Document Description	docDesc	Description of document attached.	N
Document Type	docType	Used to categorize the type of document being sent.	N
Case ID	caseId	ID for the case in which the document or comment should be associated.	N
Action Code	actionCode	Defines the code for the action that is to be performed. This should be a valid action as defined in KDD_ACTION and/or PMF. Action Code is only validated against KDD_ACTION. It is not validated that it is a valid action for the case in the current status. Action Code for external document attach is CA985. Action Code for external comment is CA986	N

Name	Parameter Name	Description	Required
Source	source	Where the document/comment came from.	N

Along with the above mentioned parameters, if document/file is to be sent, attach the document/file as multipart/form-data. For more information, see the Sample Java Code for the Service Call section.

6.4.1.4 Service Response

This service responds with a JSON object as the response data. The following is a sample response data:

{

status: “Success”,

description: “The request has been successfully processed”

}

For the possible response messages, see the Response Messages section.

6.4.1.5 Sample Java Code for the Service Call

This section shows an example of how to write a REST client in Java to call this service.

- Create a URL (java.net.URL) object and Open the Connection:

URL url = new URL(<SERVER_URL>);

HttpURLConnection urlConnection = (HttpURLConnection) url.openConnection();

> **NOTE** <SERVER_URL>: It is the end point URL for the service. For example, see the Service URL section.

- Set the request Headers:

```
urlConnection.setRequestMethod("POST");
urlConnection.setRequestProperty("Authorization",

<basicAuthEncryptedString>);

urlConnection.setRequestProperty("Content-Type", "multipart/form-data;
boundary=" + <boundaryString>);
```

> **NOTE** `<basicAuthEncryptedString>` : It is a Base64
>
> (org.apache.commons.codec.binary.Base64) encrypted credentials in String format.
>
> `<boundaryString>` : Define your boundary String.

- Write the File/Document to the URL Connection Output Stream.

```
OutputStream outputStream = urlConnection.getOutputStream();

BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(outputStream));
```

```
// Include the section to describe the file

String LINE_FEED = "\r\n";


for(int i=0; i< <FILE_PATHS>.length; i++){

File fileToUpload = new File(<FILE_PATHS>[i]); String fileName =
fileToUplod.getName();


writer.append("--" + boundaryString).append(LINE_FEED);

writer.append( "Content-Disposition: form-data; name=\"file\"; filename=\"" +
fileName + "\"").append(LINE_FEED);


writer.append(URLConnection.guessContentTypeFromName(fileName)).appen
d(LINE_FEED);

writer.append("Content-Transfer-Encoding: binary").append(LINE_FEED);
writer.append(LINE_FEED);

writer.flush();


FileInputStream inputStream = new FileInputStream(<fileToUpload>);

byte[] buffer = new byte[4096]; int bytesRead = -1;

while ((bytesRead = inputStream.read(buffer)) != -1) {
outputStream.write(buffer, 0, bytesRead);

}

outputStream.flush(); inputStream.close();


writer.append(LINE_FEED); writer.flush();

}


// Mark the end of the multipart http request


writer.write("\r\n--" + boundaryString + "--\r\n"); writer.flush();


// Close the streams outputStream.close(); writer.close();
```

NOTE `<FILE_PATHS>`: Array of File paths to be uploaded.

- Reading the Service Response

BufferedReader httpResponseReader = new BufferedReader(new
InputStreamReader(urlConnection.getInputStream()));

6.4.1.6 Changing Case Status upon Service Execution

The status of a case can be updated when a document or comment is received. This is achieved through the Action Code parameter.

- If no Action Code is defined, the action seen on the Evidence tab is Added External Document /Added External Comments and the Resulting Status is not updated.
- If the Action Code sent is only defined in KDD_ACTION and not in PMF, then the action on the Evidence tab will be what was provided in the Action Code parameter, and the Resulting Status is not updated (and the status of the case does not change).
- If the Action Code sent is defined in both KDD_ACTION and in PMF, then two records are recorded in the Audit History. One, for the receipt of the document, and the other for the case status change. Both records have the same Action name but the Resulting Status will be different.

6.4.1.7 Response Messages

The following is a list of possible Response Messages:

Scenario	Status	Description
On Success	Success	The request has been successfully processed.
Missing mandatory parameters (entityId, entityType, comments)	Failed	Missing mandatory parameters. Please make sure you send these parameters : entityId, entityType, comments.
Invalid value for a parameter (entityType, entityId/caseId)	Failed	Invalid value for the parameter. Please make sure you send valid value for : <parameter-name>.
PMF Workflow call fails	Failed	Request has been failed : Unable to start workflow.
INSERT query fails for FCC_CM_DOC_SERVICE	Failed	Request has been failed : Unable to INSERT the record to the service table.
saveCaseAttachment procedure fails	Failed	Request has been failed : Unable to INSERT the record to attachment table.
saveCaseComment procedure fails	Failed	Request has been failed : Unable to INSERT the record to comment table.
Document upload fails from AAI api (includes invalid file extension)	Failed	Request has been failed : Unable to upload selected files.

6.4.2 Configuring Entity Types

By default, the configuration for the entity type 'CASE' is defined in the system. Any further Entity Type configuration needs an entry in the table FCC_CM_DOC_SRVCE_CONF.

Column Name	Data Type	Description
V_ENTITY_TYPE	VARCHAR2(1000 CHAR)	Defines an Entity Type.
V_MANDATORY_PARAMS	VARCHAR2(4000 CHAR)	Set of parameter to be made mandatory for the Entity Type (Comma separated).
V_ALLWD_DOCUMENT_TYPE	CLOB	For Future Use.
V_ALLWD_ACTION_CD	VARCHAR2(4000 CHAR)	For Future Use.
V_ALLWD_SOURCE	CLOB	For Future Use.
V_ALLWD_USER_IDS	CLOB	For Future Use.

For example, if you want to define/configure a new entity type as "CUSTOMER" with senderId and actionCode as mandatory along the default mandatory parameters mentioned in the Request/Query Parameters section, then make the entries in `FCC_CM_DOC_SRVCE_CONF` table as mentioned below:

V_ENTITY_TYPE	V_MANDATORY_PARAMS	V_ALLWD_DOCUMENT_TYPE	V_ALLWD_ACTION_CD	V_ALLWD_SOURCE	V_ALLWD
CUSTOMER	senderId,actionCode	(null)	(null)	(null)	(null)

7 Appendix A - Supported JSON Structures for Generic Event Information

> **NOTE** This section applies only to the following API Services
>
> Create Event
>
> Create Event and Promote to Case
>
> Create Event and Extend to Existing Case

This part of the API allows you to add generic evented data to any event.

> **NOTE** The entries in these samples are only for reference purposes.

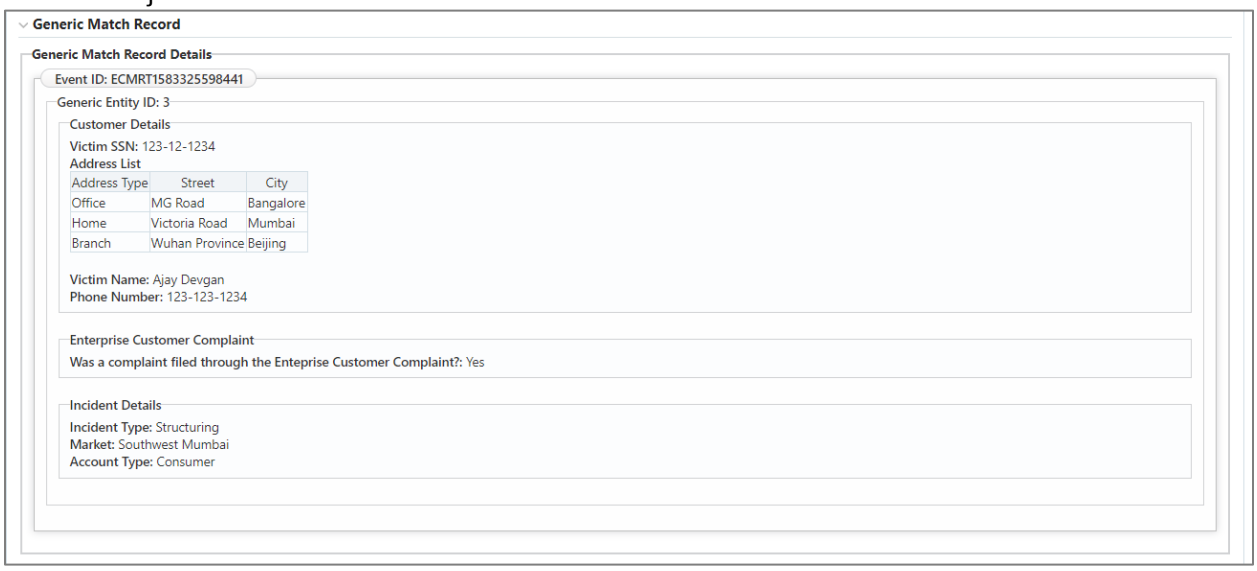
1. Simple JSON with simple Objects (One Level)

```
{
                    "Customer Details": {
                    "Victim Name": "John Doe",
                    "Victim SSN": "123-12-1234",
                      "Phone Number": "123-123-1234"
        },
        "Enterprise Customer Complaint": {
                "Was a complaint filed through the Enterprise Customer Complaint?": "Yes"
        },
        "Incident Details": {
                "Incident Type": "Structuring",
                "Account Type": "Consumer",
                "Market": "Southwest Pennsylvania"
        }
}
```

2. Simple JSON with one level of Array Object

```
{
        "Customer Details": {
            "Victim Name": "Ajay Devgan",
            "Victim SSN": "123-12-1234",
            "Phone Number": "123-123-1234",
            "Address List": [{
                    "Address Type": "Office",
                    "Street": "MG Road",
                    "City": "Bangalore"
                },
                {
                    "Address Type": "Home",
                    "Street": "Victoria Road",
                    "City": "Mumbai"
                },
                {
                    "Address Type": "Branch",
                    "Street": "Wuhan Province",
                    "City": "Beijing"
                }
            ]
        },
        "Enterprise Customer Complaint": {
```

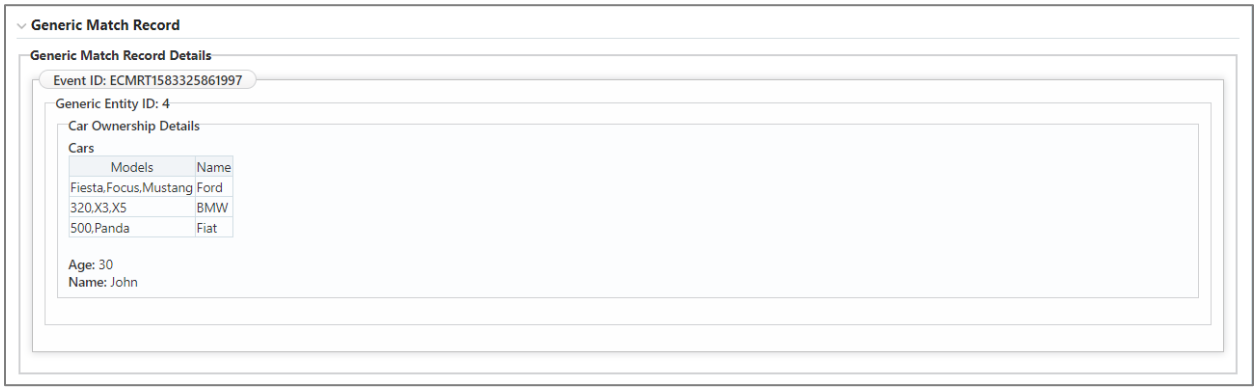
```
                "Was a complaint filed through the Enterprise Customer Complaint?":
"Yes"
            },
            "Incident Details": {
                "Incident Type": "Structuring",
                "Account Type": "Consumer",
                "Market": "Southwest Mumbai"
            }
        }
```



3. Simple Object with Two levels of Array and Second Array being simple String of Values

```
{
            "Car Ownership Details": {
                "Name": "John",
                "Age": 30,
                "Cars": [{
                        "Name": "Ford",
                        "Models": ["Fiesta", "Focus", "Mustang"]
                    },
                    {
                        "Name": "BMW",
                        "Models": ["320", "X3", "X5"]
                    },
                    {
```

```
                    "Name": "Fiat",
                    "Models": ["500", "Panda"]
                }
            ]
        }
    }
```

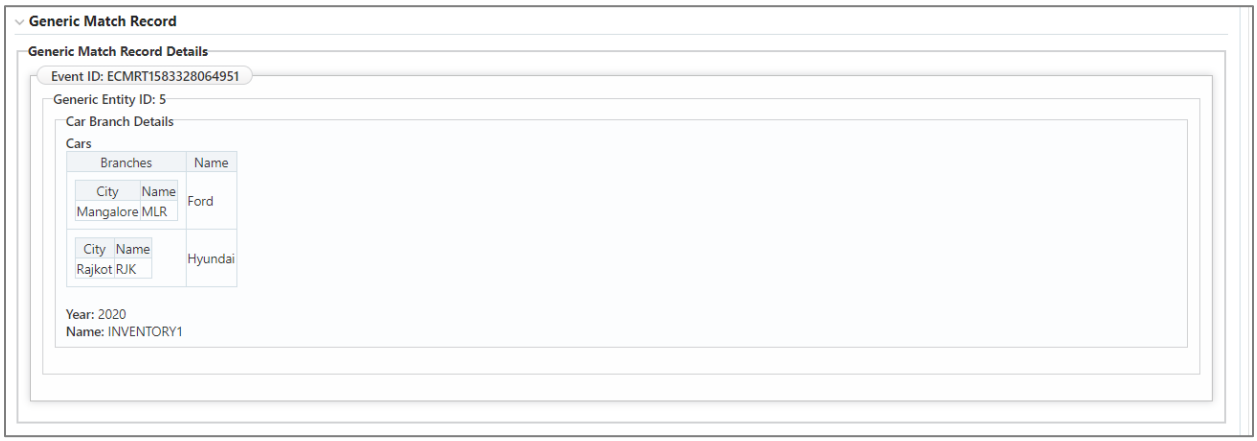


4. Simple Object with Two levels of Array and Second Array also consisting of Key Values

```
{
        "Car Branch Details": {
            "Name": "INVENTORY1",
            "Year": 2020,
            "Cars": [{
                    "Name": "Ford",
                    "Branches": [{
                            "Name": "BLR",
                            "City": "Bangalore"
                    }, {
                            "Name": "MLR",
                            "City": "Mangalore"
                    }]
                },
                {
                    "Name": "Hyundai",
                    "Branches": [{
```

```
                    "Name": "DEL",
                    "City": "Delhi"
                }, {
                    "Name": "RJK",
                    "City": "Rajkot"
                }]
            }

        ]
    }
}
```

5. Array JSON Object with two levels of Array and Second Array also consisting of key values

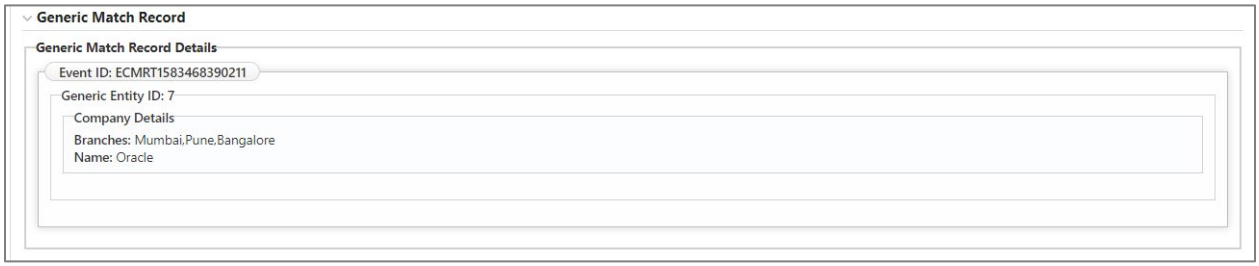

```
{
    "Bank Details": [{
        "Name": "HDFC",
        "Year": 1988,
        "Departments": [{
            "Name": "Credit",
            "Employees": [{
                "Name": "Virat",
                "City": "Bangalore"
            }, {
                "Name": "Smith",
```

```
                        "City": "Jaipur"
                    }]
                },
                {
                    "Name": "Debit",
                    "Employees": [{
                        "Name": "Ricky",
                        "City": "Bangalore"
                    }, {
                        "Name": "Sanath",
                        "City": "Kandy"
                    }]
                }
            ]
        },
        {
            "Name": "ICICI",
            "Year": 1992,
            "Departments": [{
                    "Name": "Credit",
                    "Employees": [{
                        "Name": "Sachin",
                        "City": "Mumbai"
                    }, {
                        "Name": "Kapil",
                        "City": "Delhi"
                    }]
                },
                {
                    "Name": "Debit",
                    "Employees": [{
                        "Name": "Steve",
                        "City": "Sydney"
                    }, {
```

```
                                                    "Name": "Marvan",
                                                    "City": "Colombo"
                                                }]
                                            }
                                        ]
                                    }
                                ]
                            }
```

6. Simple Object and Inside Simple Array of Strings

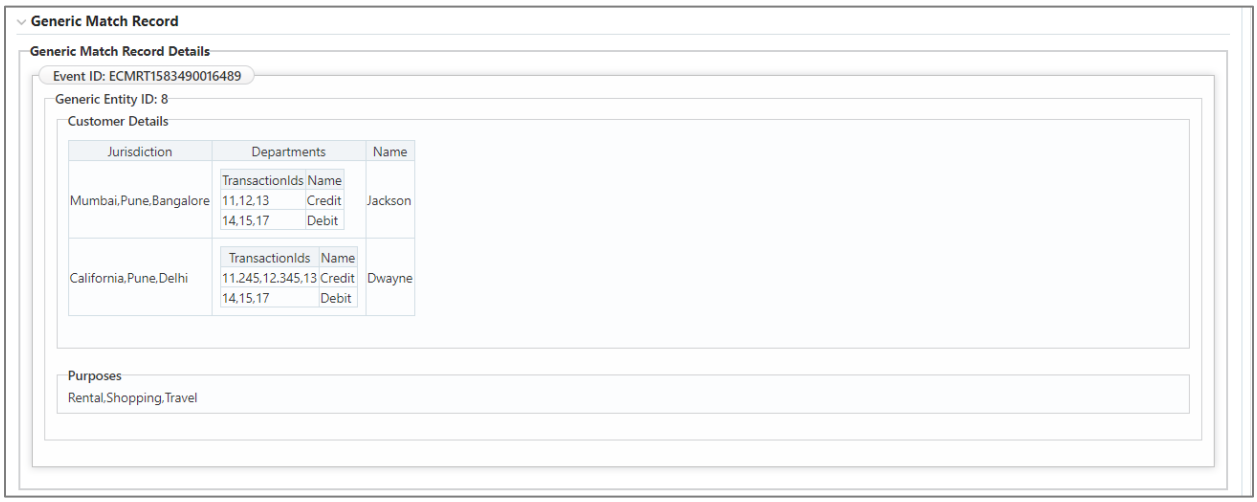
```
{
        "Company Details": {
            "Name": "Oracle",
            "Branches": ["Mumbai", "Pune", "Bangalore"]
        }
}
```



7. Array JSON Object with two levels of Array and Second Array consisting of Simple Int Values. Also, Simple Array of Strings (For Example : Purposes)

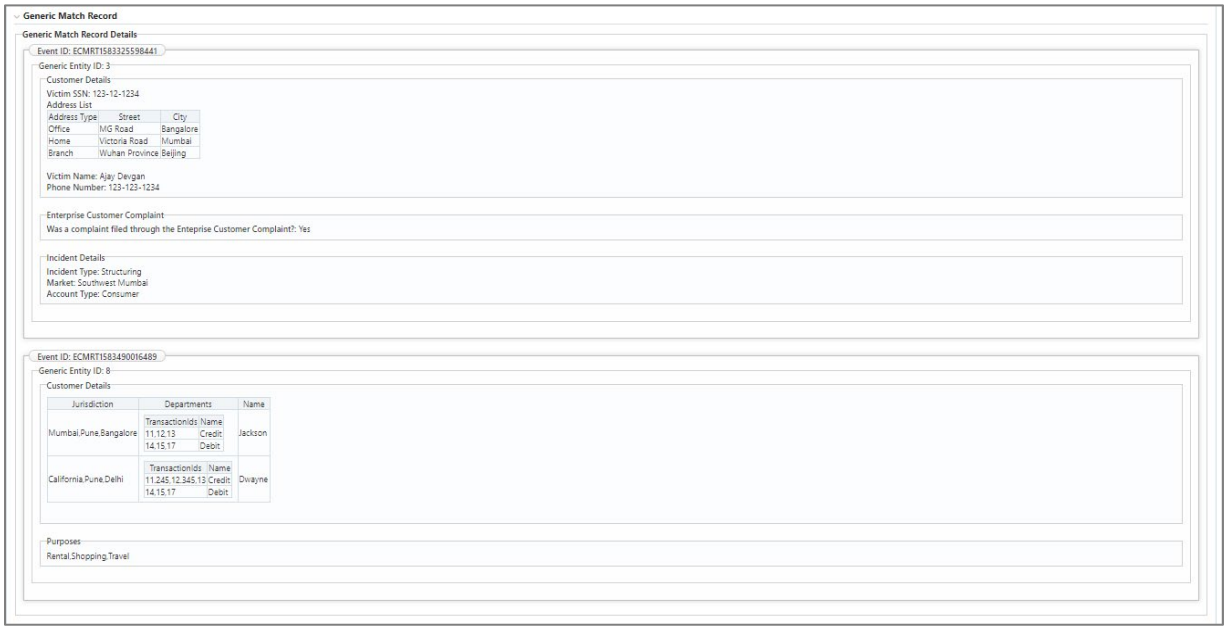
```
{
        "Customer Details": [{
                "Name": "Jackson",
                "Jurisdiction": ["Mumbai", "Pune", "Bangalore"],
                "Departments": [{
                                "Name": "Credit",
                                "TransactionIds": [11, 12, 13]
                        },
                        {
                                "Name": "Debit",
                                "TransactionIds": [14, 15, 17]
                        }
                ]
        },
        {
                "Name": "Dwayne",
                "Jurisdiction": ["California", "Pune", "Delhi"],
                "Departments": [{
                                "Name": "Credit",
                                "TransactionIds": [11.245, 12.345, 13]
                        },
                        {
                                "Name": "Debit",
                                "TransactionIds": [14, 15, 17]
                        }
                ]
```

```
            }
        ],
        "Purposes": ["Rental", "Shopping", "Travel"]
    }
```



8. More than 1 Generic Entity Under Same Event

9. Multiple Events selected on UI



7.1 Assumptions and Notes

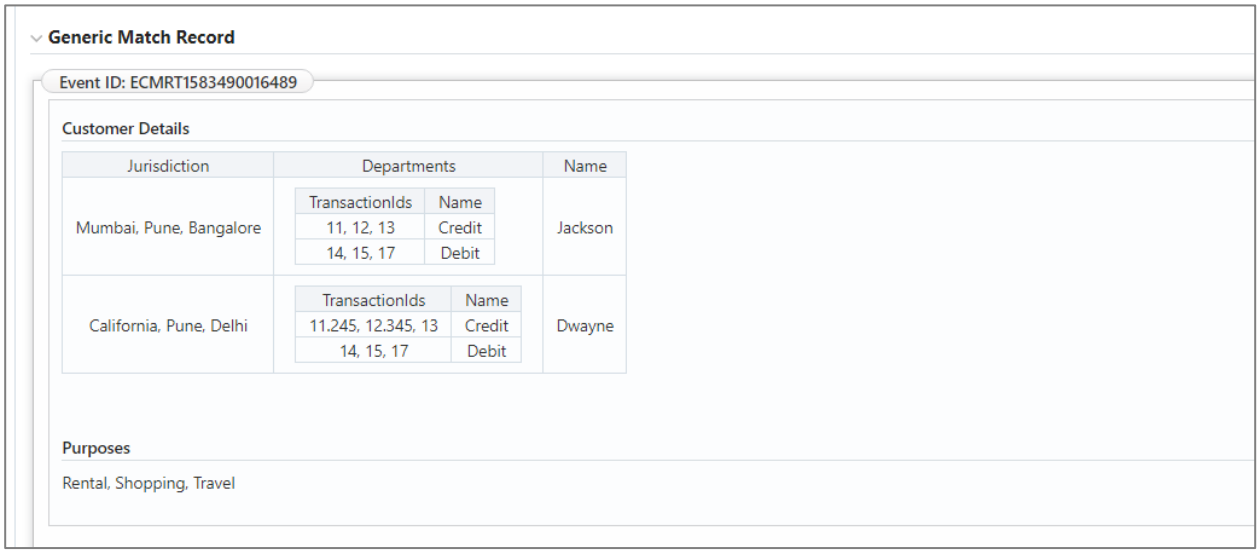
1. Section Names or Object Names must be provided for all elements which are of type Object or Array. For example, following json without section Name will not be supported:

```
{
    "Name": "ASK",
    "Work": "IT"
}
```

2. Supports 2 levels of Arrays for Simple Objects (Main JSON Object->Array1->Array2). Similarly, for the Array Object, it will be Main Array Object->Array1->Array2.

3. Supports simple object inside Array, but this is a rare case where we use any objects inside object. Use array where all single objects can pass it as Array. Else, it as good as normal key value attributes to the parent object:

```
[
    {
        "Name": "ASK",
        "Work": "IT"
    }
]
```



4. For generic entity JSON structures for the Array Type JSON which is depicted as Table on the UI, the order in which the columns are shown is random.

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to the OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the My Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site that has all the revised or recently released documents.