# Oracle Banking Digital Experience

**Development Guide for Alert Configuration**
**Release 19.1.0.0.0**

**Part No. F18558-01**

**May 2019**

ORACLE®

# Table of Contents

# 1. Preface

## 1.1 Intended Audience

This document is intended for the following audience*:*

- Customers

- Partners

## 1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

## 1.3 Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

## 1.4 Structure

This manual is organized into the following categories:

*Preface* gives information on the intended audience. It also describes the overall structure of the User Manual.

*Introduction* provides brief information on the overall functionality covered in the User Manual.

The subsequent chapters provide information on transactions covered in the User Manual.

Each transaction is explained in the following manner:

- Introduction to the transaction

- Screenshots of the transaction

- The images of screens used in this user manual are for illustrative purpose only, to provide improved understanding of the functionality; actual screens that appear in the application may vary based on selected browser, theme, and mobile devices.

- Procedure containing steps to complete the transaction- The mandatory and conditional fields of the transaction are explained in the procedure.

If a transaction contains multiple procedures, each procedure is explained. If some functionality is present in many transactions, this functionality is explained separately.

## 1.5 Related Information Sources

For more information on Oracle Banking Digital Experience Release 19.1.0.0.0, refer to the following documents:

- Oracle Banking Digital Experience Licensing Guide

- Oracle Banking Digital Experience Installation Manuals

# 2.  Context

This alert configuration contains step to configure alerts for any event in the system. Alerts are configured against the pre configured activity event .Any alert is identified by 3 properties as follows:

1.  Activity Id: An identifier for the activity being performed .It is the combination of the fully qualified name for the class and the method name.

    E.g. - Request fund activity in the Wallet.

    The activity id would be "**com.ofss.digx.app.wallet.service.core. Wallet.requestFunds**"

2.  Event Id: An identifier for the event occurred while performing the activity. An activity can have multiple events. It should start from the module name followed by the logical name for the event.

    E.g.  – Request fund success is an event in the wallet module.

    The Event Id can be "**WA_REQUEST_FUNDS_SUCCESS**"

3.  Action Id: An identifier for the action to be executed during event processing .The action can be of 3 types.

    a.  Alerts: Raise a message alert for the specified destination type like EMAIL, SMS etc. This is the default action performed while alerts processing.

    b.  Notifications: The notifications to be generated for the dashboard etc.

    c.  Business Logic: Any business Logic to be performed while alerts processing.

# 3.    Database Configurations

All the configurations are explained with respect to Wallets request fund activity.

1. The Activity entry is added in the **DIGX_EP_ACT_B** table.

```
Insert into digx_ep_act_b
  (COD_ACT_ID,
   TXT_ACT_NAME,
   TXT_ACT_DESC,
   MODULE_TYPE,
   CREATED_BY,
   CREATION_DATE,
   LAST_UPDATED_BY,
   LAST_UPDATE_DATE,
   OBJECT_VERSION_NUMBER,
   OBJECT_STATUS)
values
   ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'Wallet.requestFunds',
   'Wallet Request Funds',
   'WA',
   'SYSTELLER',
   sysdate,
   'SYSTELLER',
   sysdate,
   1,
   'A');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_ACT_ID | Primary key of the table. An identifier for the activity raising the event. It is the combination of the fully qualified name for the class and the method name. |
| TXT_ACT_NAME | Name of the activity. As a convention it is '**.**' separated combination of class name and method name. |
| TXT_ACT_DESC | Description of the activity. |
| MODULE_TYPE | Module type of the activity. It maps to the **ModuleType** enumeration. |

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_EP_ACT_B.sql

2. The event is added in the table **DIGX_PM_EVENT_ALL_B** table.

```
Insert into digx_pm_event_all_b
  (EVENT_CODE, EVENT_DESC, ALERTS_FLAG)
values
  ('WA_REQUEST_FUNDS_SUCCESS',
   'Wallet Request Funds Successful',
   'Y');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| EVENT_CODE | Primary key of the table. An identifier for the event occurred. It should start from the module type followed by the logical name for the event. |
| EVENT_DESC | Description of the event. |
| ALERTS_FLAG | Identifies whether the alert is required for this event or not. Possible values : **'Y'** or **'N'**. |

SVN Location for Seed Data Script :

http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_PM_EVENT_ALL_B.sql

3. The activity Event combination is added in **DIGX_EP_ACT_EVT_B** table. Separate entries are required for all the events of the activity i.e. Suppose activity **'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds'** has two events one for success and other for failure, 2 entries will go in the table for both of them.

```
insert into DIGX_EP_ACT_EVT_B
  (COD_ACT_ID,
   COD_EVENT_ID,
   TXT_ACT_EVT_DESC,
   TXT_EVT_TYP,
   TXT_ACT_EVT_TYP)
values
  ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'WA_REQUEST_FUNDS_SUCCESS',
   'Wallet Request Funds Successful',
   'OTHER',
   'ONLINE');
```

```
insert into DIGX_EP_ACT_EVT_B
  (COD_ACT_ID,
   COD_EVENT_ID,
   TXT_ACT_EVT_DESC,
   TXT_EVT_TYP,
   TXT_ACT_EVT_TYP)
values
  ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'WA_REQUEST_FUNDS_FAILURE',
   'Wallet Request Funds Failure',
   'OTHER',
   'ONLINE');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_ACT_ID | Activity Id. It must match to the **COD_ACT_ID** column of **DIGX_EP_ACT_B** table. |
| COD_EVENT_ID | Event Id. It must match to the **EVENT_CODE** column of **DIGX_PM_EVENT_ALL_B** table. |
| TXT_ACT_EVT_DESC | Description of the activity event combination. |
| TXT_EVT_TYP | Event type. It maps to **EventType** enumeration. |
| TXT_ACT_EVT_TYP | Activity Event type. It maps to **ActivityEventType** enumeration. Possible values : '**BULK'** or **'ONLINE'**. |

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_EP_ACT_EVT_B.sql

4. Message templates are added based on the destination types to the table
   **DIGX_EP_MSG_TMPL_B** table.

```
insert into digx_ep_msg_tmpl_b
  (COD_TMPL_ID,
   DESTINATION_TYPE,
   MSG_TMPL_NAME,
   MSG_TMPL_DESC,
   TXT_MSG_TMPL,
   CREATED_BY,
   CREATION_DATE,
   LAST_UPDATED_BY,
   LAST_UPDATE_DATE,
   OBJECT_VERSION_NUMBER,
   OBJECT_STATUS,
   TXT_SUBJECT_TMPL,
   DETERMINANT_VALUE)
values
  ('WA_REQUEST_FUNDS_EMAIL',
   'EMAIL',
   'Wallet Request Funds EMAIL',
   '',
   '<p>Dear #WalletId#.</p></br><p>Please fund my wallet by
#Amount#.</p></br></br><p>Regards</p></br>#SenderName#',
   'SYSTELLER',
   sysdate,   'SYSTELLER',
   sysdate,
   1,
   'A',
   'Fund Wallet Request',
   'OBDX_BU');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_TMPL_ID | Primary key of the table. Uniquely identifies a message template. It should start from the module type followed by the logical name for the template. |
| DESTINATION_TYPE | Destination type of the template. It maps to **DestinationType** enumeration. |
| MSG_TMPL_NAME | Logical name of the message template. |
| MSG_TMPL_DESC | Description of the message template. |
| TXT_MSG_TMPL | It contains the format for the message body. It is stored as CLOB in the table. |
| TXT_SUBJECT_TMPL | It contains the subject for the message. It is also stored as CLOB in the table. |
| DETERMINANT_VALUE | It determines the entity code for the template. |

SVN Location for Seed Data Script :

http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_EP_MSG_TMPL_B.sql

As you can see in the above example, the data elements like wallet id, amount and sender name are defined in between **'#'**. The entry for those data elements(or attributes) is done in the following tables.

5. Message attributes are added in the table **DIGX_EP_MSG_ATTR_B** table.

```
insert into digx_ep_msg_attr_b
  (COD_MESS_TMPL_ID, COD_ATTR_ID, ATTR_MASK, DETERMINANT_VALUE)
values
  ('WA_REQUEST_FUNDS_EMAIL', 'Amount', 'D', 'OBDX_BU');
```

```
insert into digx_ep_msg_attr_b
  (COD_MESS_TMPL_ID, COD_ATTR_ID, ATTR_MASK, DETERMINANT_VALUE)
values
  ('WA_REQUEST_FUNDS_EMAIL', 'WalletId', 'D','OBDX_BU');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_MESS_TMPL_ID | Message template Id. It must match to the **COD_TMPL_ID** column of **DIGX_EP_MSG_TMPL_B** table. |
| COD_ATTR_ID | Name of the attribute. It must match to the one defined inside **TXT_MSG_TMPL** of **DIGX_EP_MSG_TMPL_B** table. |
| ATTR_MASK | Masking format for the attribute value. Characters given as **'X'** will be masked and the ones given as **'D'** will be displayed as it is. |
| DETERMINANT_VALUE | It determines the entity code for the template. |

SVN Location for Seed Data Script :

http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/metadata/DIGX_MD_SERVICE_ATTR.sql

6. Service attributes are added in **DIGX_MD_SERVICE_ATTR** table.

```
insert into digx_md_service_attr
  (COD_SERVICE_ATTR_ID,
   TYP_DATA_AVAIL,
   TYP_DATA_SRC,
   COD_ATTR_ID,
   COD_SERVICE_ID,
   PARAMETER_NAME,
   CREATED_BY,    CREATION_DATE,
   LAST_UPDATED_BY,    LAST_UPDATE_DATE,
   OBJECT_VERSION_NUMBER,
   OBJECT_STATUS,
   REF_FIELD_DEFN_ID)
values
 ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds.WalletId.INPUT',
   'INDIRECT',
   'INPUT',
   'WalletId',
   'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'requestFundDTO',
   'SETUP',    sysdate,
   'SETUP',    sysdate,
   1,
   'A',
   'com.ofss.digx.app.wallet.dto.transfer.RequestFundDTO.WalletId.Value');
```

```
insert into digx_md_service_attr
  (COD_SERVICE_ATTR_ID,
   TYP_DATA_AVAIL,
   TYP_DATA_SRC,
   COD_ATTR_ID,
   COD_SERVICE_ID,
   PARAMETER_NAME,
   CREATED_BY,    CREATION_DATE,
   LAST_UPDATED_BY,    LAST_UPDATE_DATE,
   OBJECT_VERSION_NUMBER,
   OBJECT_STATUS,
   REF_FIELD_DEFN_ID)
values
  ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds.Amount.INPUT',
   'INDIRECT',
   'INPUT',
   'Amount',
   'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'requestFundDTO',
   'SETUP',    sysdate,
   'SETUP',    sysdate,
   1,
   'A',
   'com.ofss.digx.app.wallet.dto.transfer.RequestFundDTO.Amount.Amount');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_SERVICE_ATTR_ID | Primary key of the table. As a convention, **'.'** separated **c**ombination of **COD_SERVICE_ID**, **COD_ATTR_ID** and **TYP_DATA_SRC**. |
| TYP_DATA_AVAIL | Possible values : **'DIRECT'** or **'INDIRECT'**. **'DIRECT'** only when **'TYP_DATA_SRC'** is **'INPUT'** and the attribute value is one of the arguments passed to the activity. Otherwise **'INDIRECT'**. |
| TYP_DATA_SRC | Possible values : **'INPUT'** or **'DTO'**. **'INPUT'** when the attribute value can be obtained from the arguments passed to the activity. **'DTO'** when the attribute value cannot be obtained from the arguments and is genertated/fetched within the activity. |
| COD_ATTR_ID | Name of the attribute. |
| COD_SERVICE_ID | Activity Id. It must match to **COD_ACT_ID** column of **DIGX_EP_ACT_B** table. |
| PARAMETER_NAME | The name of the argument passed to the activity. Its value will be null when **TYP_DATA_SRC** is **'DTO'**. |
| REF_FIELD_DEFN_ID | Fully qualified path from which the attribute value can be obtained. |

Here , in case of TYP_DATA_SRC as 'INPUT', there can be 2 cases :

- The attribute value is passed directly to the activity i.e. the attribute value is one of the arguments passed to the activity. In this case, TYP_DATA_AVAIL will be 'DIRECT'.

- The attribute value is not passed directly to the activity , but it can be obtained from one of the arguments passed to the activity i.e. it is part of one of the DTOs which is passed to the activity. In this case, TYP_DATA_AVAIL will be 'INDIRECT'.

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/metadata/DIGX_MD_SERVICE_ATTR.sql

7. Source of the message attributes are added in **DIGX_EP_MSG_SRC_B** table.

```
INSERT INTO digx_ep_msg_src_b
  (COD_MESS_TMPL_ID, COD_ATTR_ID, COD_ACT_ID, COD_SERVICE_ATTR_ID,
DETERMINANT_VALUE)
VALUES
 ('WA_REQUEST_FUNDS_EMAIL',
  'WalletId',
  'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
  'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds.WalletId.INPUT',

  'OBDX_BU');
```

```
INSERT INTO digx_ep_msg_src_b
  (COD_MESS_TMPL_ID, COD_ATTR_ID, COD_ACT_ID, COD_SERVICE_ATTR_ID,
DETERMINANT_VALUE)
VALUES
  ('WA_REQUEST_FUNDS_EMAIL',
   'Amount',
   'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds.Amount.INPUT',
   'OBDX_BU');
```

| COLUMN NAME | DESCRIPTION |
| --- | --- |
| COD_MESS_TMPL_ID | Message template Id. It must match to the **COD_TMPL_ID** column of **DIGX_EP_MSG_TMPL_B** table. |
| COD_ATTR_ID | Name of the attribute. It must match to the one defined inside **TXT_MSG_TMPL** of **DIGX_EP_MSG_TMPL_B** table. |
| COD_ACT_ID | Activity Id. It must match to **COD_ACT_ID** column of **DIGX_EP_ACT_B** table. |
| COD_SERVICE_ATTR_ID | Service attribute id. It must match to **COD_SERVICE_ATTR_ID** of **DIGX_MD_SERVICE_ATTR** table. |
| DETERMINANT_VALUE | It determines the entity code for the template. |

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_EP_MSG_SRC_B.sql

8. The attributes which are input to the activity are added in **DIGX_MD_SERVICE_INPUTS** table.

```
insert into digx_md_service_inputs
  (COD_SERVICE_ID,
   PARAMETER_NAME,
   PARAMETER_INDEX,
   DATA_TYPE,
   CREATED_BY,
   CREATION_DATE,
   LAST_UPDATED_BY,
   LAST_UPDATE_DATE,
   OBJECT_VERSION_NUMBER,
   OBJECT_STATUS)
values
  ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'requestFundDTO',
   1,
   'com.ofss.digx.app.wallet.dto.transfer.RequestFundDTO',
   'SETUP',
   sysdate,
   'SETUP',
   sysdate,
   1,
   'A');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_SERVICE_ID | Activity Id. It must match to **COD_ACT_ID** column of **DIGX_EP_ACT_B** table. |
| PARAMETER_NAME | The name of the argument passed to the activity. |
| PARAMETER_INDEX | Unique index of the argument for an activity. It starts from 0 for a particular activity. |
| DATA_TYPE | Data type of the argument passed to the activity. |

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_EP_MSG_SRC_B.sql

9. The generic attributes along with their datatypes are added in
   **DIGX_MD_GEN_ATTR_LEGACY_B** table.

```sql
insert into DIGX_MD_GEN_ATTR_LEGACY_B

  (cod_constraint_attr_id,
   txt_constraint_attr_name,
   data_type,
   CREATED_BY,
   CREATION_DATE,
   LAST_UPDATED_BY,
   LAST_UPDATE_DATE,
   OBJECT_VERSION_NUMBER,
   OBJECT_STATUS)
values
  ('WALLET_ID',
   'UniqueWalletIdentifier',
   'java.lang.String',
   'SETUP',
   sysdate,
   'SETUP',
   sysdate,
   1,
   'A');
```

```sql
insert into DIGX_MD_GEN_ATTR_LEGACY_B
  (cod_constraint_attr_id,
   txt_constraint_attr_name,
   data_type,
   CREATED_BY,
   CREATION_DATE,
   LAST_UPDATED_BY,
   LAST_UPDATE_DATE,
   OBJECT_VERSION_NUMBER,
   OBJECT_STATUS)
values
  ('AMOUNT',
   'TransactionAmount',
   'java.lang.String',
   'SETUP',
   sysdate,
   'SETUP',
   sysdate,
   1,
   'A');
```

| COLUMN NAME | DESCRIPTION |
| --- | --- |
| COD_CONSTRAINT_ATTR_ID | Attribute Id. |
| TXT_CONSTRAINT_ATTR_NAME | Name or description of the attribute. |
| DATA_TYPE | Data type of the attribute to format the attribute value. |

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/metadata/DIGX_MD_GEN_ATTR_LEGACY_B.sql

10. Entry for alert is added in **DIGX_EP_ACT_EVT_ACN_B** table.

```
insert into digx_ep_act_evt_Acn_b
   (COD_ACT_ID,
    COD_EVENT_ID,
    COD_ACTION_ID,
    FLG_TRANSACTIONAL,
    COD_DEC_ID,
    FLG_CONDITIONAL,
    COD_ACN_TMPL_ID,
    ALERT_NAME,
    CREATED_BY,
    CREATION_DATE,
    LAST_UPDATED_BY,
    LAST_UPDATE_DATE,
    OBJECT_VERSION_NUMBER,
    EXPIRY_DATE,
    ALERT_TYPE,
    ALERT_DISPATCH_TYPE,
    OBJECT_STATUS)
values
   ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
    'WA_REQUEST_FUNDS_SUCCESS',
    'A',
    'Y',
    '0',
    'N',
    '1',
    'Wallet Request Funds',
    'OFSSUser',
    sysdate,
    'OFSSUser',
    sysdate,
    1,
    to_date('23-02-2018', 'dd-mm-yyyy'),
    'S',
    'I',
    'A');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_ACT_ID | Activity Id. |
| COD_EVENT_ID | Event Id. |
| COD_ACTION_ID | Action Id. Possible value : **'A'** (means Alert) |
| FLG_TRANSACTIONAL | Possible values : 'Y' or 'N'. This flag indicates whether events under this event category are transactional events or not.A Transactional event is an event which get processed within the same session of manager API. |
| COD_DEC_ID | Possible Value : 0 |
| FLG_CONDITIONAL | Possible value : 'N'. |

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_ACN_TMPL_ID | Possible values : **1** or **2** . 1 indicates the importance of alert is critical. 2 indicates the importance of alert is informational. |
| ALERT_NAME | Unique name for the alert. |
| EXPIRY_DATE | Expiry Date of the alert. |
| ALERT_TYPE | Alert Type. Possible values: **'M'** or **'S'**. 'M' indicates the alert is of mandatory type and cannot be subscribed/unsubscribed by the user. 'S' indicates the alert is of subscribed type which can be subscribed/unsubscribed by the user. |
| ALERT_DISPATCH_TYPE | Alert Dispatch Type. Possible values: **'I'** or **'D'**. 'I' indicates immediate i.e. the alert needs to be send immediately. 'D' indicates deffered i.e. the alert will be sent later. |

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_EP_ACT_EVT_ACN_B.sql

11. Entry for recipient message templates is added in **DIGX_EP_EVT_REC_B** table. Separate entries are required for all the destination types of the alert i.e. Suppose activity **'com.ofss.digx.app.wallet.service.core.Wallet.requestFunds'** has two destination types, EMAIL and SMS, 2 entries will go in this table.

```
insert into digx_ep_evt_rec_b
  (COD_ACT_ID,
   COD_EVENT_ID,
   COD_ACTION_ID,
   COD_MSG_TMPL_ID,
   TXT_DEST_TYP,
   SUBSCRIBER_TYPE,
   SUBSCRIBER_VALUE,
   ALERT_TYPE,
   LOCALE)
values
  ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'WA_REQUEST_FUNDS_SUCCESS',
   'A',
   'WA_RequestFunds_EMAIL',
   'EMAIL',
   'PARTY',
   'CUSTOMER',
   'S',
   'en');
```

```
insert into digx_ep_evt_rec_b
  (COD_ACT_ID,
   COD_EVENT_ID,
   COD_ACTION_ID,
   COD_MSG_TMPL_ID,
   TXT_DEST_TYP,
   SUBSCRIBER_TYPE,
   SUBSCRIBER_VALUE,
   ALERT_TYPE,
   LOCALE)
values
  ('com.ofss.digx.app.wallet.service.core.Wallet.requestFunds',
   'WA_REQUEST_FUNDS_SUCCESS',
   'A',
   'WA_RequestFunds_SMS',
   'SMS',
   'PARTY',
   'CUSTOMER',
   'S',
   'en');
```

| COLUMN NAME | DESCRIPTION |
|---|---|
| COD_ACT_ID | Activity Id. |
| COD_EVENT_ID | Event Id. |
| COD_ACTION_ID | Action Id. Possible value : **'A'** (means Alert) |
| COD_MSG_TMPL_ID | Message Template Id. Foreign key to **COD_TMPL_ID** of **DIGX_EP_MSG_TMPL_B.** |
| TXT_DEST_TYP | Destination Type. Possible value : **'EMAIL'** , **'SMS',** **'SECURE_MAIL_BOX' , 'PUSH_NOTIFICATION'** |
| 'SUBSCRIBER_TYPE | Possible value : **'PARTY'.** |
| SUBSCRIBER_VALUE | Possible value : **'CUSTOMER'.** |
| ALERT_TYPE | Alert Type. Possible values: **'M'** or **'S'**. 'M' indicates the alert is of mandatory type and cannot be subscribed/unsubscribed by the user. 'S' indicates the alert is of subscribed type which can be subscribed/unsubscribed by the user. |
| LOCALE | Locale to pick the location/language specific template for. |

SVN Location for Seed Data Script :
http://obcpsvn.oraclecorp.com:8080/svn/clip/trunk/core/seed/oracle/alerts/DIGX_EP_EVT_REC_B.sql

**Note:** Entries for most of the activities, events, corresponding activity events , message templates, message attributes are already added. Please check for the entries in the table to avoid repetition.

## 3.1 API for Raising an EVENT

For raising an event, registerActivityAndGenerateEvent API has been provided in the **AbstractApplication** class.

It takes 4 parameters:

- Session Context
- EventId
- ActivityLog

Alerts can be either Account based or Party based.

If it is Account based, populating 2 attributes(accountId and accountType) of ActivityLog is bare minimum requirement.

Similarily, if it is Party based,populating 1 attribute(customerId) of ActivityLog is bare minimum requirement.

For the other attributes, In case the attribute is already present in **com.ofss.digx.app.alerts.dto.eventgen.ActivityLog** class use the existing ActivityLog instance.

Else create a subclass of ActivityLog having your attribute. Set the attribute value in the ActivityLog child class and pass its instance as an argument to r**egisterActivityAndGenerateEvent** method.

```
ActivityLog activityLog = new ActivityLog();
// if it is a party based alert setCustomerId
        activityLog.setCustomerId(sessionContext.getTransactingPartyCode());
// if it is a account based alert setAccountId and setAccountType
        activityLog.setAccountId("<<AccountNumber>>");
        activityLog.setAccountType("<<AccountType>>");
//If required, set other attributes in activityLog

super.registerActivityAndGenerateEvent(sessionContext, <<EventId>>, activityLog);
```

## 3.2 Custom Fields For Push notifications

Following Keys can be used to customize Push Notifications.

| KEY NAME | VALUE |
|---|---|
| SOUND_IOS | File name of custom sound file added to OBDX IOS App |
| SOUND_ANDROID | File name of custom sound file added to OBDX Android App |
| LARGE_ICON_ANDROID | URL of icon image to be displayed as large icon in Big Style Push Notification of OBDX Android App. |
| LARGE_IMAGE_ANDROID | URL of image to be displayed in Big Style Push Notification of OBDX Android App. |

These custom keys are to be added to the value of "TXT_MSG_TMPL" column of **DIGX_EP_MSG_TMPL_B** table.

If alerts are being created through front end UI, add following keys to "Notification Message" section.

Syntax for adding custom keys to Push Notification alert messages

[customfield1Name~customfield1Value|customfield2Name~customfield2Value]

**Example 1:**

You have requested for #NoOfChequeBook# cheque book with #ChequeBookOption# leaves on Account #AccountNo#.

[SOUND_ANDROID~isntit|LARGE_IMAGE_ANDROID~http://static1.squarespace.com/static/54a c6f9ae4b0cf1d82a4b59e/t/587f9e52cd0f68e84c5548fd/1484758653422/?format=300w|SOUND_I OS~chime.m4a]

**Example 2:**

You have requested for #NoOfChequeBook# cheque book with #ChequeBookOption# leaves on Account #AccountNo#.

[SOUND_ANDROID~isntit|LARGE_ICON_ANDROID~http://static1.squarespace.com/static/54ac 6f9ae4b0cf1d82a4b59e/t/587f9e52cd0f68e84c5548fd/1484758653422/?format=300w|SOUND_I OS~chime.m4a]

## 3.3    Multi-Entity Specific templates

Entity specific templates can be created by following ways :

1.  Creation of a new alert and template before the entity creation.
    If a new alert has to be maintained before the creation of any new entity, the data for the same has to be inserted in the following tables twice.
    One for DETERMINANT_VALUE '*' and the other for DETERMINANT_VALUE 'OBDX_BU', which is the default entity.
    Tables :
    **DIGX_EP_MSG_TMPL_B**
    **DIGX_EP_MSG_ATTR_B**
    **DIGX_EP_MSG_SRC_B**

2.  Creation of a new alert and template after the entity creation.
    If a new alert has to be maintained after the creation of entity/entities, the same can be replicated for the different entities using the below queries.
    First insert the templates for DETERMINANT_VALUE '*' and DETERMINAT_VALUE 'OBDX_BU' and then execute the below queries for the respective entities.

```
insert into DIGX_EP_MSG_TMPL_B (DETERMINANT_VALUE, CREATION_DATE,
LAST_UPDATED_DATE, OBJECT_VERSION_NUMBER, COD_TMPL_ID, DESTINATION_TYPE,
MSG_TMPL_NAME, MSG_TMPL_DESC, TXT_MSG_TMPL, CREATED_BY, LAST_UPDATED_BY,
OBJECT_STATUS, TXT_SUBJECT_TMPL, DOMAIN_OBJECT_EXTN)
    (select '<ENTITY_CODE_TO_BE_REPLICATED>', sysdate, sysdate, 1,
COD_TMPL_ID, DESTINATION_TYPE, MSG_TMPL_NAME, MSG_TMPL_DESC, TXT_MSG_TMPL,
CREATED_BY, LAST_UPDATED_BY, OBJECT_STATUS, TXT_SUBJECT_TMPL,
DOMAIN_OBJECT_EXTN  from DIGX_EP_MSG_TMPL_B where DETERMINANT_VALUE = '*');

insert into DIGX_EP_MSG_ATTR_B (DETERMINANT_VALUE, COD_MESS_TMPL_ID,
COD_ATTR_ID, ATTR_MASK, DATA_ATTR_ORDER, DOMAIN_OBJECT_EXTN)
    (select '<ENTITY_CODE_TO_BE_REPLICATED>', COD_MESS_TMPL_ID, COD_ATTR_ID,
ATTR_MASK, DATA_ATTR_ORDER, DOMAIN_OBJECT_EXTN  from DIGX_EP_MSG_ATTR_B where
DETERMINANT_VALUE = '*');

insert into DIGX_EP_MSG_SRC_B (DETERMINANT_VALUE, COD_MESS_TMPL_ID,
COD_ATTR_ID, COD_ACT_ID, COD_SERVICE_ATTR_ID)
    (select '<ENTITY_CODE_TO_BE_REPLICATED>', COD_MESS_TMPL_ID, COD_ATTR_ID,
COD_ACT_ID, COD_SERVICE_ATTR_ID  from DIGX_EP_MSG_SRC_B where
DETERMINANT_VALUE = '*');
```