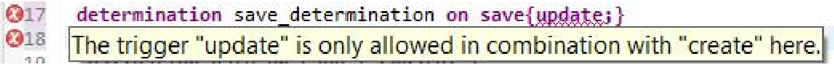
RAP: Determination

Determination:

- Used to determine the value of certain entity based on the trigger condition.

Note:

- The execution order of determinations is not fixed. If there is more than one determination triggered by the same condition, the execution order is arbitrary.
- Once a determination has been triggered, it must run independently from other determinations.
- In unmanaged scenarios, determinations are only supported for draft instances, not for active instances.
- For determinations defined as on save, update works only in combination with the trigger operation create.



```
determination save_determination on save{update;}
```

The trigger "update" is only allowed in combination with "create" here.

Trigger conditions:

- Create
- Update
- Delete
- Field modification

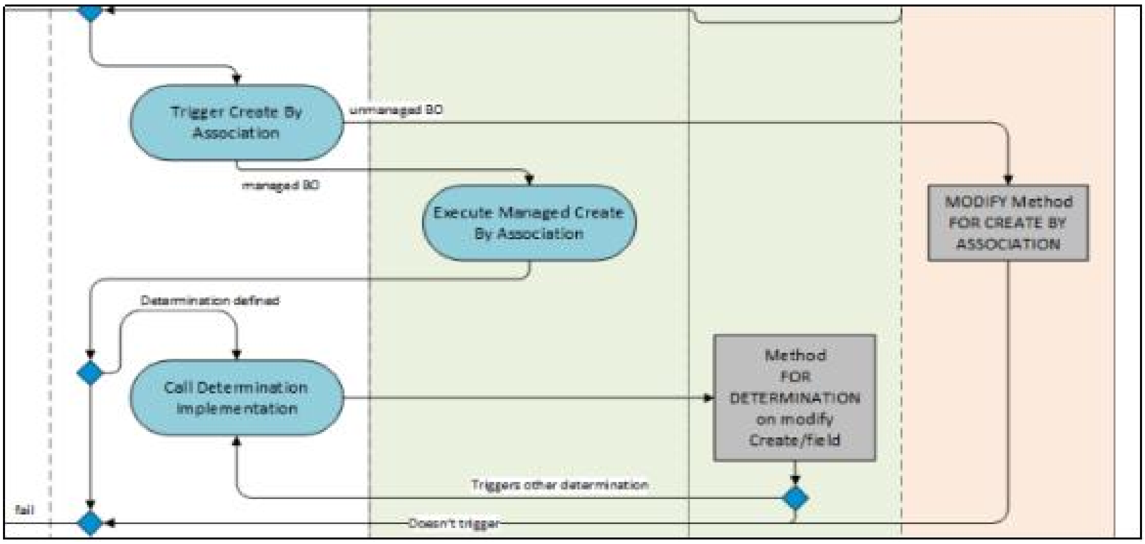
Good practices:

- Don't change the field that is being triggered, as it could cause an infinite loop. The trigger field and the calculation field should always be different. For example, if you use 'field1' as the trigger field and then update the same field within the method, it will trigger the method again, resulting in an endless loop.
- The trigger conditions can contain only fields of that entity. It cannot contain fields of child / parent entity.
- It is recommended to use either a trigger condition as a standard operation or a field to avoid multiple calls. For example, if you have a trigger condition set to 'update' and 'field1,' every time you update the field, it will trigger the determination twice because you are satisying both conditions.

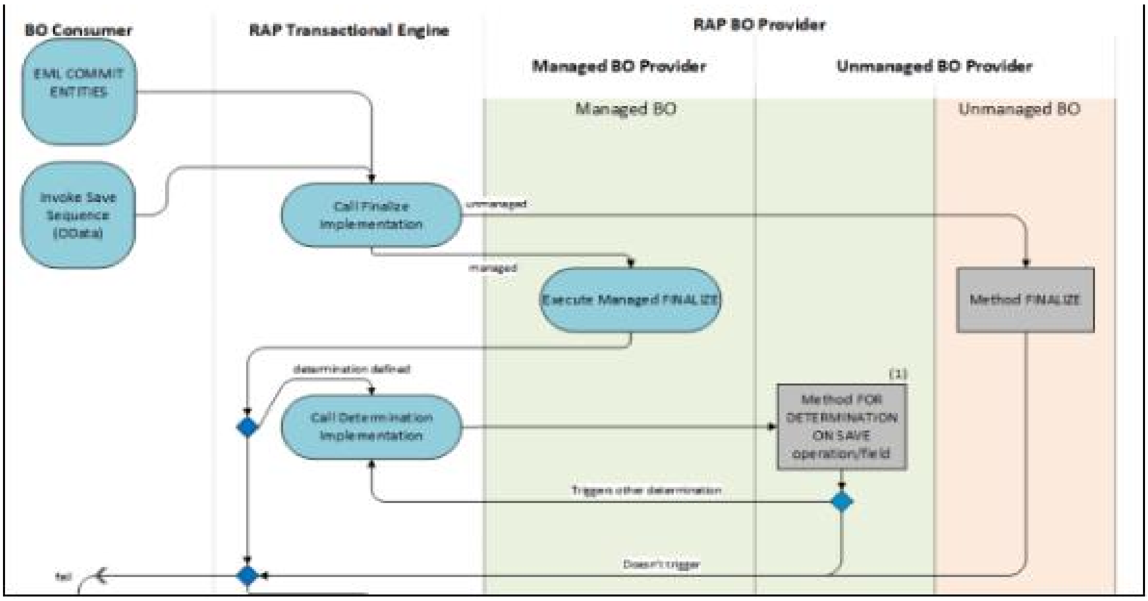
Differnce between determination on modify and on save:

On modify	On save
Executed just after the modification of transactional buffer.	Executed before persisting on table.
Triggered before save and on save as well.	Triggered only at save
Executed in interactive phase	Executed in save sequence.

Flow of determination:


Trigger Create By Association
unmanaged BO
managed BO
Execute Managed Create By Association
MODIFY Method FOR CREATE BY ASSOCIATION
Determination defined
Call Determination Implementation
Method FOR DETERMINATION on modify Create/field
Triggers other determination
fail
Doesn't trigger

Determination on modify - Interative phase



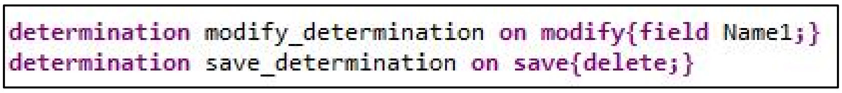
Determination on save - Save sequence

Determination on save and finalize:

- **Determination on save is triggerd only on managed scenerio.**
- **In unmanged scenerio, finalize method will be triggered, instead of determination on save method**

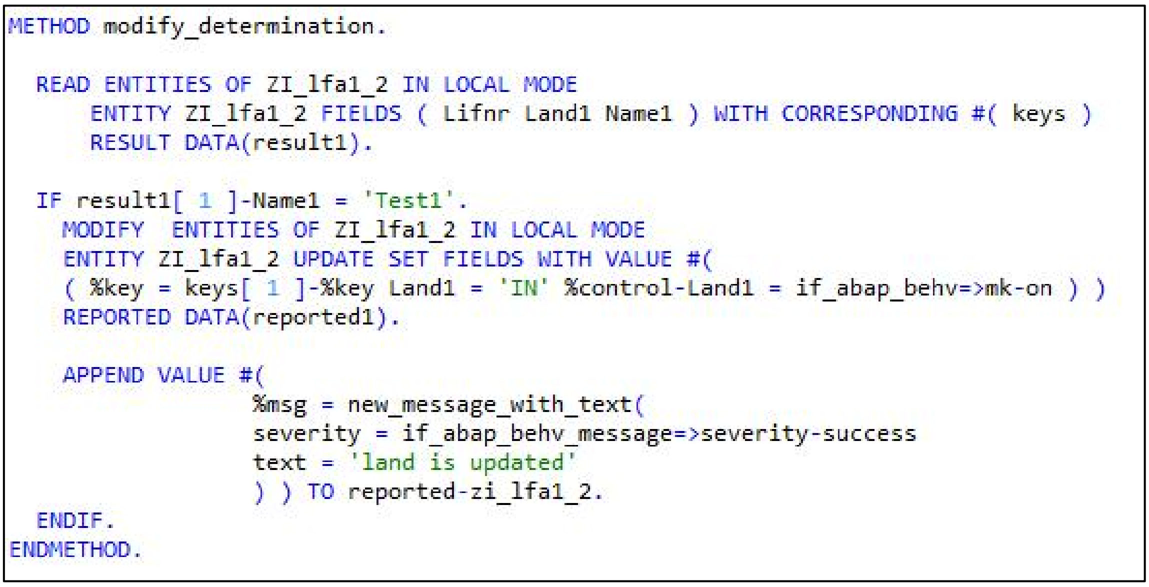
Steps to develop determination:

1. Add determination in bdef interface with trigger condition.



```
determination modify_determination on modify{field Name1;}
determination save_determination on save{delete;}
```

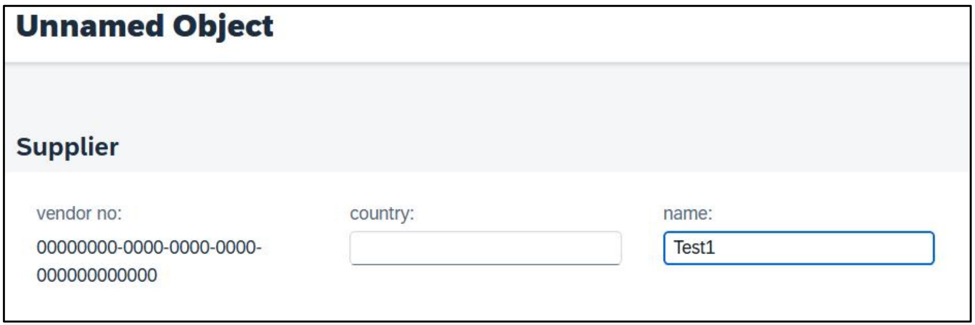
2. Implement the logic for determnation in abap behaviour pool.



```
METHOD modify_determination.

  READ ENTITIES OF ZI_lfa1_2 IN LOCAL MODE
      ENTITY ZI_lfa1_2 FIELDS ( Lifnr Land1 Name1 ) WITH CORRESPONDING #( keys )
      RESULT DATA(result1).

  IF result1[ 1 ]-Name1 = 'Test1'.
    MODIFY  ENTITIES OF ZI_lfa1_2 IN LOCAL MODE
    ENTITY ZI_lfa1_2 UPDATE SET FIELDS WITH VALUE #(
    ( %key = keys[ 1 ]-%key Land1 = 'IN' %control-Land1 = if_abap_behv=>mk-on ) )
    REPORTED DATA(reported1).

    APPEND VALUE #(
                  %msg = new_message_with_text(
                  severity = if_abap_behv_message=>severity-success
                  text = 'land is updated'
                  ) ) TO reported-zi_lfa1_2.
  ENDIF.
ENDMETHOD.
```

Requirement:

If the user provides name as “Test1”, Then the country should be “IN”.



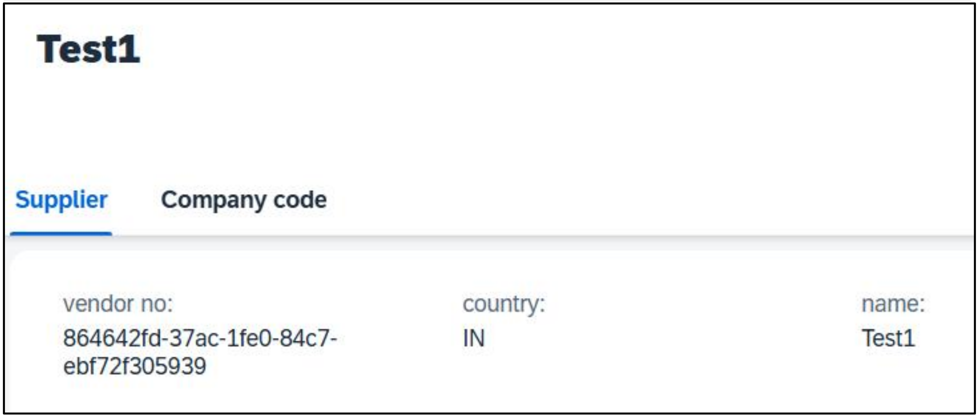
Unnamed Object

Supplier

vendor no:
00000000-0000-0000-0000-000000000000

country:

name:
Test1



Test1

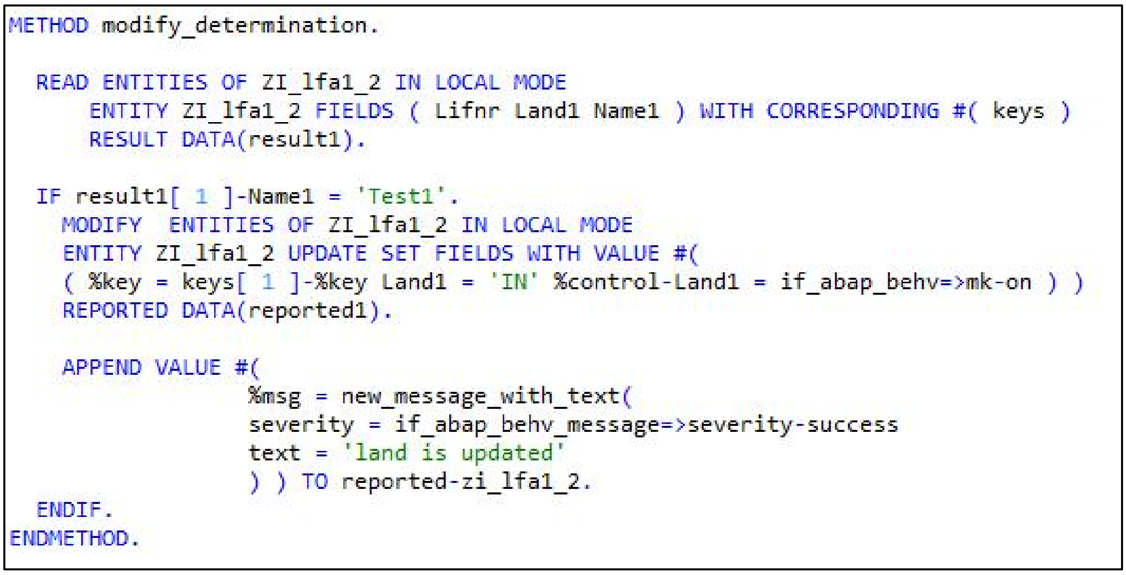
Supplier Company code

vendor no:
864642fd-37ac-1fe0-84c7-ebf72f305939

country:
IN

name:
Test1

To achieve this, we have declared the determination on modify with the field name. Since the determination is triggered after the changes are moved to the transactional buffer, we can use the 'read EML' to read the data based on the key’s importing parameter. If the name is 'Test1,' we modify the country to 'IN' using the 'modify EML' in the transactional buffer.



```
METHOD modify_determination.

  READ ENTITIES OF ZI_lfa1_2 IN LOCAL MODE
      ENTITY ZI_lfa1_2 FIELDS ( Lifnr Land1 Name1 ) WITH CORRESPONDING #( keys )
      RESULT DATA(result1).

  IF result1[ 1 ]-Name1 = 'Test1'.
    MODIFY  ENTITIES OF ZI_lfa1_2 IN LOCAL MODE
    ENTITY ZI_lfa1_2 UPDATE SET FIELDS WITH VALUE #(
    ( %key = keys[ 1 ]-%key Land1 = 'IN' %control-Land1 = if_abap_behv=>mk-on ) )
    REPORTED DATA(reported1).

    APPEND VALUE #(
                   %msg = new_message_with_text(
                   severity = if_abap_behv_message=>severity-success
                   text = 'land is updated'
                   ) ) TO reported-zi_lfa1_2.
  ENDIF.
ENDMETHOD.
```