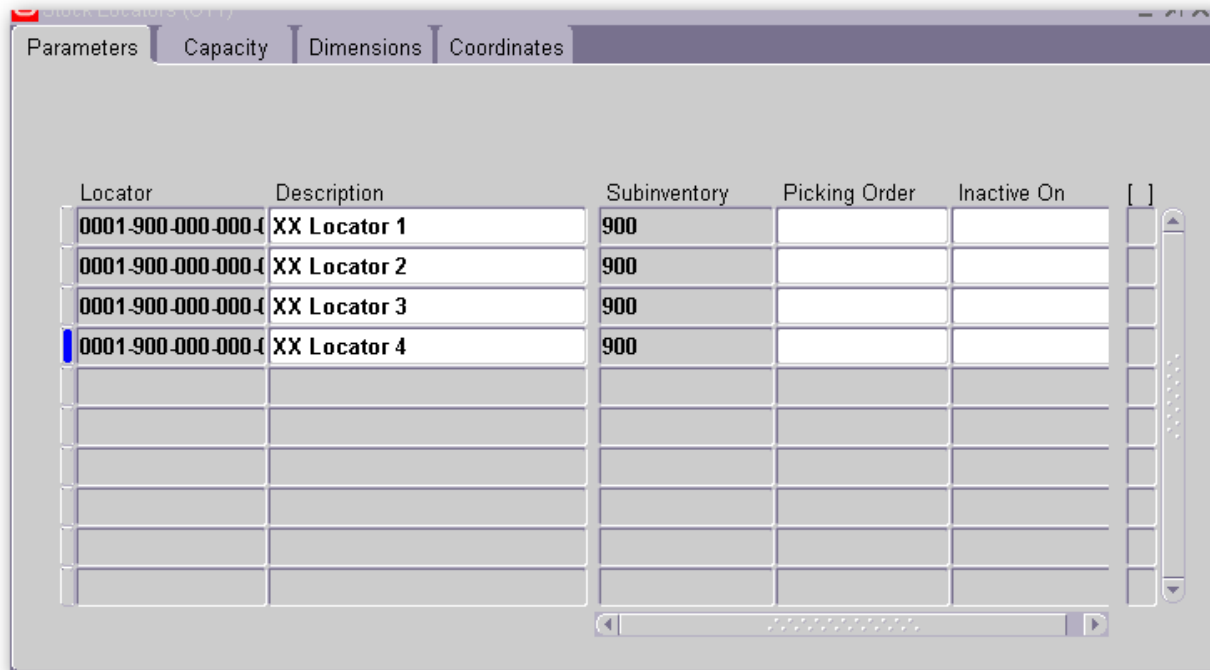


How to Delete Stock Locators by Using API

1. Locators

If user wants to delete some inactive locators but in oracle application delete button is disabled on locators interface. But Oracle provided an API which can be used for deletion of locators. Here I took following locators for demonstration.



Locator	Description	Subinventory	Picking Order	Inactive On
0001-900-000-000-1	XX Locator 1	900		
0001-900-000-000-1	XX Locator 2	900		
0001-900-000-000-1	XX Locator 3	900		
0001-900-000-000-1	XX Locator 4	900		

2. Create Table

Create a new table by using following script with all locator segments

```
CREATE TABLE DEV_DATA
(
  SR_NO          NUMBER,
  INV_ORG_ID     NUMBER,
  INVENTORY_ORGANIZATION NUMBER,
  SUB_INVENTORY CHAR(3 BYTE),
  WAREHOUSE      CHAR(3 BYTE),
  FLOR           CHAR(3 BYTE),
  ROWN           CHAR(3 BYTE),
  RACK           CHAR(3 BYTE),
  BIN            CHAR(3 BYTE),
  BOX            CHAR(3 BYTE),
  SUPPLIER       CHAR(4 BYTE),
  SPARE1         CHAR(3 BYTE),
  SPARE2         CHAR(3 BYTE),
  CODE           CHAR(45 BYTE),
  CREATION_DATE  DATE,
```

```
STATUS          CHAR(1 BYTE)
)
```

3. Upload Data in Table by Using SQLLDR

Control File Syntax

```
LOAD DATA
  INFILE 'Data.csv'
  APPEND INTO TABLE Dev_Data
  REPLACE FIELDS TERMINATED BY ','
  TRAILING NULLCOLS
(
  SR_NO "trim(:SR_NO)",
  INV_ORG_ID "trim(:INV_ORG_ID)",
  INVENTORY_ORGANIZATION "trim(:INVENTORY_ORGANIZATION)",
  SUB_INVENTORY "trim(:SUB_INVENTORY)",
  WAREHOUSE "trim(:WAREHOUSE)",
  FLOR "trim(:FLOR)",
  ROWN "trim(:ROWN)",
  RACK "trim(:RACK)",
  BIN "trim(:BIN)",
  BOX "trim(:BOX)",
  SUPPLIER "trim(:SUPPLIER)",
  SPARE1 "trim(:SPARE1)",
  SPARE2 "trim(:SPARE2)",
  CODE "trim(:CODE)",
  CREATION_DATE "trim(:CREATION_DATE)"
)
```

Verify Data is uploaded successfully by using following two queries

```
Select count(*) from dev_data
```

```
Select * from dev_data
```

4. Inactive Locator By Passing Inactive date

5. Deleting Script (API)

Connect by Using APPS Credentials to Database
Run the following scripts this will delete locator

```
DECLARE
l_msg_data          VARCHAR2 (100);
l_msg_count         NUMBER;
l_return_status    VARCHAR2 (1);
l_locator_id       NUMBER;
l_locator_exists   VARCHAR2 (1);
l_org_id           NUMBER          := 116;  /*Organization_id */
l_organization_code VARCHAR2 (10)  := 'U11'; /*Organization_Code */
l_sub_code        VARCHAR2 (10);  /*Variable for Subinventory*/
l_concatenated_segments VARCHAR2 (100); /*Variable for Locator Segment*/
l_user_id         NUMBER          := 1262;
                  /* User ID From FND_users Table */
l_resp_id         NUMBER          := 20634; /*Responsibility Id*/
```

```

l_resp_appl_id      NUMBER      := 401;
                    /* Responsibility Application id */

/*Fetch data into cursor for deletion of Locator*/

CURSOR c_loc
IS
  SELECT inventory_location_id, organization_id, description,
         subinventory_code sub_inventory, code
  FROM apps.dev_data;

v_loc              c_loc%ROWTYPE;
BEGIN

/* APPS_INITIALIZE required because indirectly use profile options */

  fnd_global.apps_initialize (l_user_id, l_resp_id, l_resp_appl_id);

--Open Cursor

  OPEN c_loc;

  LOOP
    FETCH c_loc
    INTO v_loc;

    EXIT WHEN c_loc%NOTFOUND;
    l_concatenated_segments := v_loc.code;
    l_sub_code := v_loc.sub_inventory;
    l_locator_id := v_loc.inventory_location_id;

    fnd_msg_pub.initialize;

    DBMS_OUTPUT.put_line ( 'Trying to Delete '
                          || l_concatenated_segments
                          || ':'
                          || l_locator_id
                          );
    inv_loc_wms_pub.delete_locator
      (x_return_status      => l_return_status,
       x_msg_count          => l_msg_count,
       x_msg_data          => l_msg_data,
       p_inventory_location_id  => l_locator_id,
       p_organization_id     => l_org_id,
       p_organization_code    => 'U11',
       p_concatenated_segments => l_concatenated_segments,
       p_validation_req_flag  => 'N'
      );

    COMMIT;
    DBMS_OUTPUT.put_line ('Return Status ' || l_return_status);

    IF l_return_status IN ('E', 'U')
    THEN
      DBMS_OUTPUT.put_line ('# of Errors ' || l_msg_count);

```

```
IF l_msg_count = 1
THEN
  DBMS_OUTPUT.put_line ('Error ' || l_msg_data);
ELSE
  FOR i IN 1 .. l_msg_count
  LOOP
    DBMS_OUTPUT.put_line ('Error ' || fnd_msg_pub.get (i, 'F'));
  END LOOP;
END IF;
ELSE
  DBMS_OUTPUT.put_line ('Locator Id is ' || l_locator_id);
END IF;
END LOOP;
CLOSE c_loc;
END;
```

6. Check values