

```

DECLARE
/*Creates the customer and customer profile Data*/
  custAccountRec      HZ_CUST_ACCOUNT_V2PUB.CUST_ACCOUNT_REC_TYPE;
  custProfileRec      HZ_CUSTOMER_PROFILE_V2PUB.CUSTOMER_PROFILE_REC_TYPE;
  custOrgRec          HZ_PARTY_V2PUB.ORGANIZATION_REC_TYPE;      --Organization
  Record Type
  custPartyRec        HZ_PARTY_V2PUB.PARTY_REC_TYPE;              --Party Record
  Type

  x_cust_account_id   NUMBER;
  x_account_number    VARCHAR2(30);
  x_party_id          NUMBER;
  x_party_number      VARCHAR2(30);
  x_profile_id        NUMBER;
  x_return_status     VARCHAR2(3);
  x_msg_count         NUMBER;
  x_msg_data          VARCHAR2(3000);

  l_collector_id      NUMBER;
BEGIN

  DBMS_OUTPUT.PUT_LINE('In createCustomer Procedure.');
```

mo_global.init('AR');

```

  fnd_global.apps_initialize ( user_id      => 1117      --user_id
                             ,resp_id      => 50677     --responsibility id
                             ,resp_appl_id => 222);     --resp application
                             id

  mo_global.set_policy_context('S',99); --99 is the org_id that I am
  using for the organization

  SELECT  collector_id
        INTO  l_collector_id
        FROM  ar_collectors
        WHERE name = 'Arun Raj';

  FOR newCustRec IN (SELECT * FROM xxraj_cust_prof_tbl --custom table
                    that holds customer and customer profile details
                    WHERE  status = 'N')

  LOOP
  /*Loop through the new customers and create the customer accounts ,
  profile, customer site and customer siste uses*/

  --Values assigned to the Customer Account Record Type
  custAccountRec.account_number      := newCustRec.account_number;
  custAccountRec.attribute_category  := newCustRec.cust_att_category;
  custAccountRec.attribute1          := newCustRec.attribute1;
  custAccountRec.attribute2          := newCustRec.attribute2;
  custAccountRec.attribute3          := newCustRec.attribute3;
  custAccountRec.attribute4          := newCustRec.attribute4;
  custAccountRec.attribute5          := newCustRec.attribute5;
  custAccountRec.attribute15         := newCustRec.cust_account_id;
  custAccountRec.orig_system_reference := newCustRec.
  orig_system_reference;
  custAccountRec.customer_class_code := newCustRec.
  customer_class_code;
  custAccountRec.status               := 'A';  --Status is Active
  custAccountRec.customer_type        := 'R';  --Customer Type
  External
  custAccountRec.created_by_module    := 'TCA_V2_API';

  --Values assinged to the Party Record Type.
```

```

custPartyRec.orig_system_reference      := newCustRec.
orig_system_reference;
custPartyRec.status                     := 'A';
custPartyRec.attribute1                  := newCustRec.attribute1;
custPartyRec.attribute2                  := newCustRec.attribute2;
custPartyRec.attribute3                  := newCustRec.attribute3;
custPartyRec.attribute4                  := newCustRec.attribute4;
custPartyRec.attribute5                  := newCustRec.attribute5;

--Values assinged to the Organization Record Type. Party Type =
Organization
custOrgRec.organization_name             := newCustRec.party_name;
custOrgRec.created_by_module             := 'TCA_V2_API';
custOrgRec.party_rec                     := custPartyRec;

/*Get Profile Details as well*/
custProfileRec.profile_class_id          := newCustRec.
profile_class_id;
custProfileRec.collector_id              := newCustRec.
collector_id;
custProfileRec.credit_checking           := newCustRec.
credit_checking;
custProfileRec.tolerance                  := newCustRec.tolerance;
custProfileRec.discount_terms            := newCustRec.
discount_terms;
custProfileRec.dunning_letters           := newCustRec.
dunning_letters;
custProfileRec.interest_charges          := newCustRec.
interest_charges;
custProfileRec.send_statements           := newCustRec.
send_statements;
custProfileRec.credit_balance_statements := newCustRec.
credit_balance_statements;
custProfileRec.credit_hold               := newCustRec.
credit_hold;
custProfileRec.override_terms            := newCustRec.
override_terms;
custProfileRec.payment_grace_days        := newCustRec.
payment_grace_days;
custProfileRec.late_charge_calculation_trx := newCustRec.
late_charge_calculation_trx;
custProfileRec.credit_items_flag         := newCustRec.
credit_items_flag;
custProfileRec.disputed_transactions_flag := newCustRec.
disputed_transactions_flag;
custProfileRec.late_charge_type          := newCustRec.
late_charge_type;
custProfileRec.interest_calculation_period := newCustRec.
interest_calculation_period;
custProfileRec.hold_charged_invoices_flag := newCustRec.
hold_charged_invoices_flag;
custProfileRec.multiple_interest_rates_flag := newCustRec.
multiple_interest_rates_flag;

/*API to create customer account, party and customer profile*/
HZ_CUST_ACCOUNT_V2PUB.CREATE_CUST_ACCOUNT (
  p_init_msg_list      => FND_API.G_TRUE,
  p_cust_account_rec   => custAccountRec,
  p_organization_rec   => custOrgRec,
  p_customer_profile_rec => custProfileRec,
  p_create_profile_amt => FND_API.G_TRUE,
  x_cust_account_id    => x_cust_account_id,

```

```

        x_account_number => x_account_number,
        x_party_id       => x_party_id,
        x_party_number   => x_party_number,
        x_profile_id     => x_profile_id,
        x_return_status  => x_return_status,
        x_msg_count      => x_msg_count,
        x_msg_data       => x_msg_data
    );

    IF x_return_status = fnd_api.g_ret_sts_success THEN
        COMMIT;
        DBMS_OUTPUT.PUT_LINE('Creation of Party of Type Organization and
        customer account is Successful ');
        DBMS_OUTPUT.PUT_LINE('x_cust_account_id : '||x_cust_account_id);
        DBMS_OUTPUT.PUT_LINE('x_account_number : '||x_account_number);
        DBMS_OUTPUT.PUT_LINE('x_party_id       : '||x_party_id);
        DBMS_OUTPUT.PUT_LINE('x_party_number : '||x_party_number);
        DBMS_OUTPUT.PUT_LINE('x_profile_id    : '||x_profile_id);
    ELSE
        DBMS_OUTPUT.put_line ('Creation of Party of Type Organization and
        customer account failed:'||x_msg_data);
        ROLLBACK;
        FOR i IN 1 .. x_msg_count
        LOOP
            x_msg_data := oe_msg_pub.get( p_msg_index => i, p_encoded =>
            'F');
            DBMS_OUTPUT.PUT_LINE( i||' ' ) '|| x_msg_data);
        END LOOP;
    END IF;
END LOOP;
commit;
END;
```