# Appendix A

## Resource CD

The resource CD accompanying this thesis contains the complete oracle developer source code with documentation for all demonstration applications and the framework. In addition, the resource CD provides binary prerequisites necessary for building and executing the demonstration applications.

## A.1 Requirements

In order to compile, deploy and run the demonstration applications, the following technologies and tools must be supported:

- Oracle 8i enterprise edition or higher version installed in the server and client machine .
- Oracle Developer 6 Installed in the client machine .

## A.2 CD Layout

## ResourceCD

### bin

- win32 Binary Prerequisites for Windows Operating Systems
- doc Thesis Paper, Abstract in English and Arabic

### src

- Center AB Application Source Code

# Appendix B

## B.1 User Interface

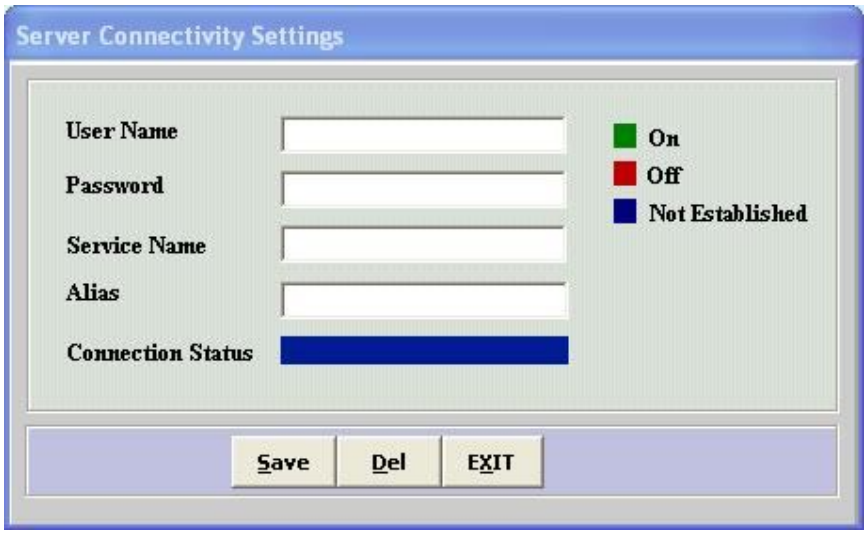
### B.1.1 Login Screen


We use this screen to login the system by providing user name and password.

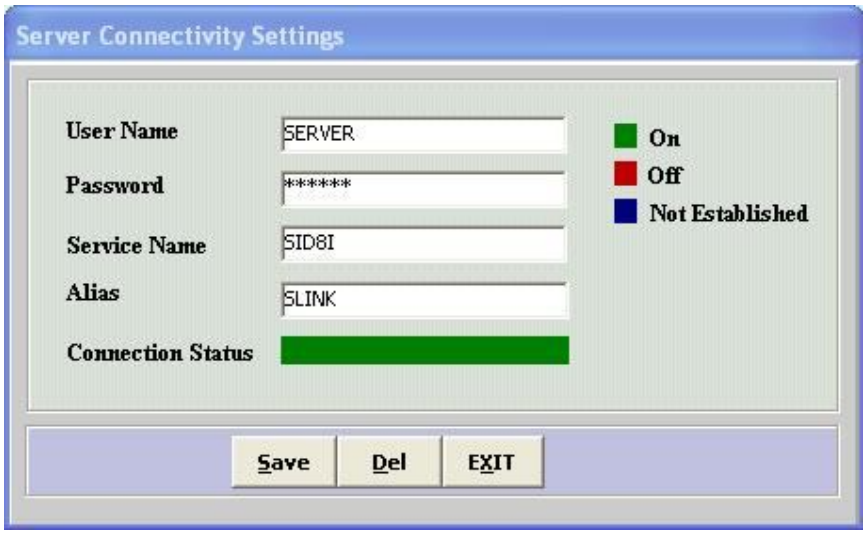
B.1.2 Update Propagation Settings


In thesis screen we determine the propagation approach the application will use to propagate changes to the server .

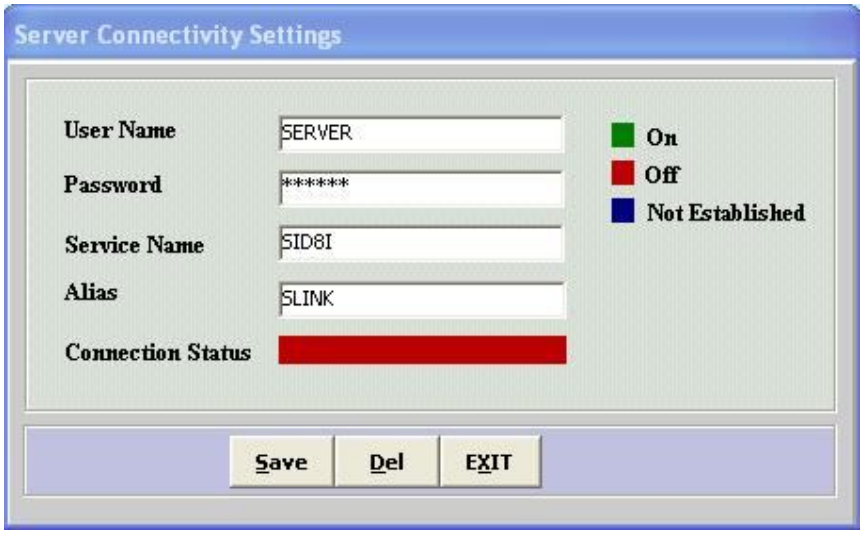
## B.1.3 Server Connectivity Settings




In this screen we provide the connection parameters of the remote server which are ***Username***,***Password*** of the remote user and the ***Service Name*** of the Remote database and the ***Alias*** the used to connect to the remote database. The colors appear in this screen provide the user or the administrator with the connection status.

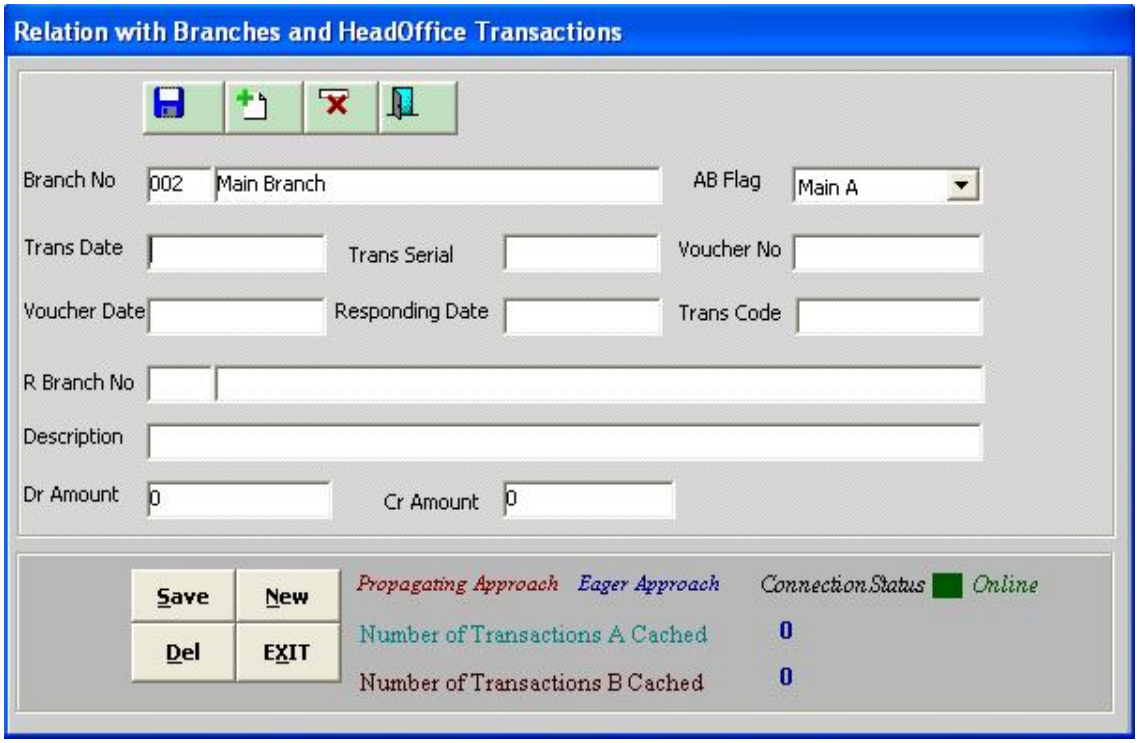



Setting screen when connection is **ON**.



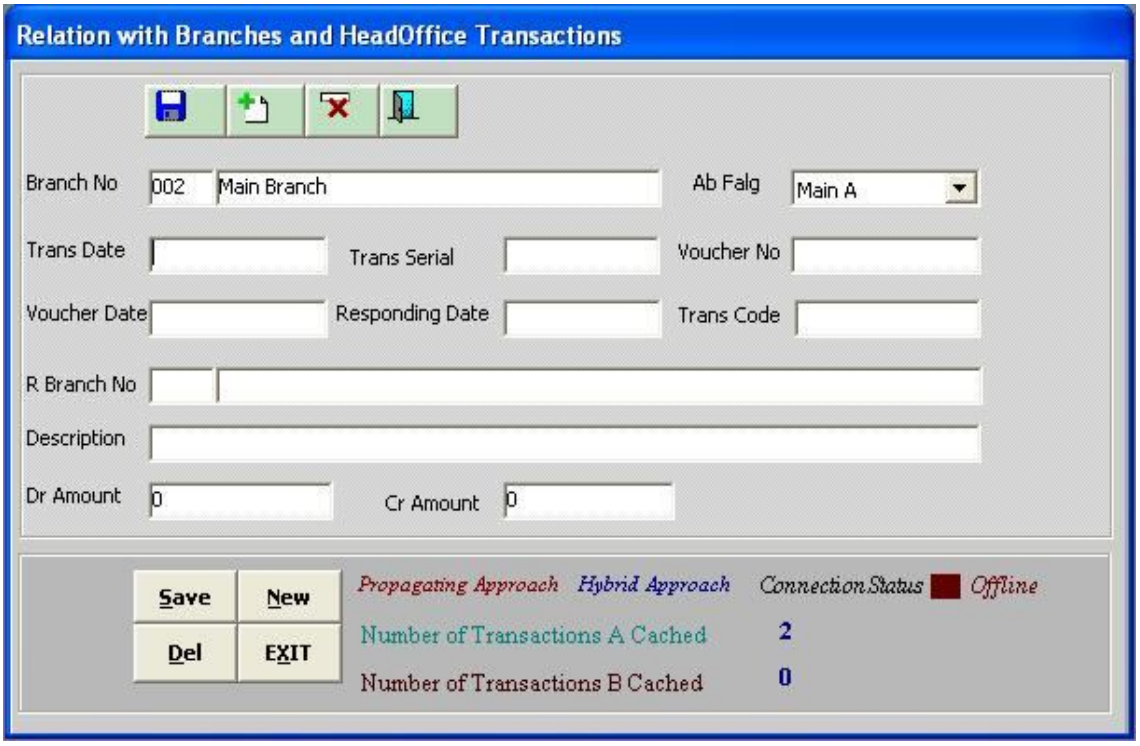

Setting Screen when connection is **OFF**

### B.1.4 Transaction Screen Connection Online




This screen is used to process main center A&B daily transactions .When running the application the system will detect the status of the network and the initialization settings of the system and provide the user ,as seen in the above screen, with the connection status and the propagation approach and the number of transactions that have been cached during offline mode.

## B.1.5 Transaction Screen Connection Offline




The screen as it appears when connection status is down .

B.1.6 Transaction Screen Connection Offline with Eager Approach:


As seen in the above screen the connection status is offline and the propagation approach as the user entered is eager . when the user enter a transaction and try to commit the application will prohibit commit changes as eager approach require online status with the server , the application interact with the user through the above message and ask the user if he desires to switch to lazy approach to commit the changes locally.

## B.2 Code

## STORED PROCEDURES

### FUNCTION IsOnline

```
Create or replace FUNCTION IsOnline RETURN Boolean IS
      cnt number;
      vlink varchar2(30);
BEGIN
    select count(*) into cnt from dblinks;
    if cnt>0 then
    begin
    select dblink into vlink from dblinks;
     EXECUTE IMMEDIATE
     'select count(*) from branches'||'@'||vlink into cnt;
      return(True);
     EXCEPTION WHEN others  THEN
   begin
       return(False);
   end;
   end;
  else
     return(false);
  end if;
END;
/
```

### Procedure Transfer_Data

```
Create or replace Procedure Transfer_Data
(
BRANCHNO                    VARCHAR2,
ABFALG                      VARCHAR2,
TDATE                       DATE,
TSERIAL                     NUMBER,
VNO                         VARCHAR2,
VDATE                       DATE,
RESPONDINGDATE              DATE,
TCODE                       VARCHAR2,
RBRANCHNO                   VARCHAR2,
DESCR2                      VARCHAR2,
DRAMOUNT                    NUMBER,
CRAMOUNT                    NUMBER,
SENDDATETIME2               DATE,
SENDFLAG                    NUMBER,
PFLAG                       VARCHAR2
) IS
```

```
VLINK           VARCHAR2(30);

BEGIN
     SELECT DBLINK INTO VLINK FROM DBLINKS;
     EXECUTE IMMEDIATE
     'Insert Into
AB_Collection'||vlink||'Values(:1,:2,:3,:4,:5,:6,:7,:8,:9,:10,:11,:12,:13,:14,:15)'
USING  BRANCHNO, ABFALG,TDATE ,
TSERIAL,VNO,VDATE,RESPONDINGDATE,TCODE,RBRANCHNO,
DESCR2,DRAMOUNT,CRAMOUNT,SENDDATETIME2,SENDFLAG,PFLAG;
END;
/
```

## Procedure Lazy_Background

```
Create or replace Procedure Lazy_Background IS
VLINK           VARCHAR2(30);
BEGIN
     SELECT DBLINK INTO VLINK FROM DBLINKS;
     update ab_daily_trans set SENDDATETIME=sysdate;
     EXECUTE IMMEDIATE
     'Insert Into AB_Collection'||vlink||'(select * from ab_daily_trans  where
send_flag=0)' ;
     update ab_daily_trans set send_flag=1 where send_flag=0;
END;
/
```

# Form Dblinks

## WHEN-NEW-FORM-INSTANCE

```
declare
         cnt number;
begin
  select count(*) into cnt from dblinks;
  if cnt>0 then
    execute_query;
    if not IsOnLine then
      set_item_property('STCOLOR',BACKGROUND_COLOR,'r190g0b0');
    end if;
  else
     set_item_property('STCOLOR',BACKGROUND_COLOR,'r0g30b150');
  end if;
end;
```

BUTTON-NEW

```
declare
          cnt number;
begin
if :uname is not null then
           if :username  =:Uname  and
      :PassWord  =:Pword  and
      :DbLink    =:DLink    and
      :Alias     =:Alyas     then
       commit_form;
            else
   commit_form;
   forms_ddl('drop database link'||' '||:DLink);
   forms_ddl('create database link'||' '||:DbLink||' '||
   'CONNECT TO'||' '||:UserName||' '||'IDENTIFIED BY'||' '||:PassWord||' '||
   'USING'||' '||""||:Alias||"");

   end if;
else
   commit_form;
   :uname:=:username;
   :pword:=:PassWord;
   :DLink:=:DbLink;
   :Alyas:=:Alias;
   forms_ddl('create database link'||' '||:DbLink||' '||
   'CONNECT TO '||' '||:UserName||' '||'IDENTIFIED BY'||' '||:PassWord||' '||
   'USING'||' '||""||:Alias||"");
end if;
  select count(*) into cnt from dblinks;
  if cnt>0 then
     if not IsOnLine then
        set_item_property('STCOLOR',BACKGROUND_COLOR,'r190g0b0');
     else
     set_item_property('STCOLOR',BACKGROUND_COLOR,'r0g100b0');
     end if;
  else
       set_item_property('STCOLOR',BACKGROUND_COLOR,'r0g30b150');
  end if;
end;
```

DELBUT

```
declare
          but number(2);
begin
  but:=show_alert('del');
  if but=alert_button1 then
     forms_ddl('drop database link'||' '||:DLink);
```

```
    go_block('DBLinks');
    delete_record;
    commit;
    clear_form;
    set_item_property('STCOLOR',BACKGROUND_COLOR,'r0g30b150');
  end if;
end;
```

## Form Trans

### WHEN-NEW-FORM-INSTANCE

```
declare
          pf number(1);
          timer_id Timer;
  Five_Secods NUMBER(5) := 5000;
          onl Boolean;
begin
timer_id := CREATE_TIMER('emp_timer', Five_Secods, REPEAT);
if IsOnLine then
       set_item_property('STCOLOR',BACKGROUND_COLOR,'r0g88b0');
       set_item_property('STdescr',FOREGROUND_COLOR ,'r0g88b0');
     :StDescr:='Online';
else
    set_item_property('STCOLOR',BACKGROUND_COLOR,'r100g0b0');
    set_item_property('STdescr',FOREGROUND_COLOR ,'r100g0b0');
    :StDescr:='Offline';
end if;
select count(*) into :ANUM from ab_daily_trans where Send_Flag=0 and
AB_FALG='A';
select count(*) into :BNUM from ab_daily_trans where Send_Flag=0 and
AB_FALG='B';

select branch_no into :branch_no from settings;
select branch_name into :braname1 from branches where branch_no=:branch_no;
select count(*) into pf from settings;
if pf>0 then
  select PropFlag into pf from settings;
  if pf=1 then
          :PRobDescr:='Eager Approach';
  elsif pf=2 then
          :PRobDescr:='Lazy Approach';
  elsif pf=3 then
    :PRobDescr:='Hybrid Approach';
  else
    :PRobDescr:='Unknown Approach';
  end if;
end if;
go_item('AB_FALG');
```

```
end;
```

## WHEN-TIMER-EXPIRED

```
declare
        pf number(1);
                onl Boolean;
                but number(2);
                cnt number;
                bo boolean;
                dblnk  varchar2(30);
begin
bo:=IsOnLine;
if bo then
      set_item_property('STCOLOR',BACKGROUND_COLOR,'r0g88b0');
      set_item_property('STdescr',FOREGROUND_COLOR ,'r0g88b0');
    :StDescr:='Online';
else
    set_item_property('STCOLOR',BACKGROUND_COLOR,'r100g0b0');
    set_item_property('STdescr',FOREGROUND_COLOR ,'r100g0b0');
    :StDescr:='Offline';
end if;
select count(*) into :ANUM from ab_daily_trans where Send_Flag=0 and
AB_FALG='A';
select count(*) into :BNUM from ab_daily_trans where Send_Flag=0 and
AB_FALG='B';


select count(*) into pf from settings;
if pf>0 then
  select PropFlag into pf from settings;
  if pf=1 then
          :PRobDescr:='Eager Approach';
  elsif pf=2 then
          :PRobDescr:='Lazy Approach';
  elsif pf=3 then
    :PRobDescr:='Hybrid Approach';
  else
    :PRobDescr:='Unknown Approach';
  end if;
end if;
end;
```

## SAVE BUTTON

```
declare
        PFlag Number(1);
        but number(2);
        OnOf Boolean;
begin
```

```
select PropFlag into PFlag from Settings;
if PFlag Is Not Null then
  OnOf:=IsOnline;
  If PFlag=1 and OnOf=True then
        :Send_Flag:=1;
        Transfer_Data
        (
          :BRANCH_NO,
          :AB_FALG,
          :T_DATE,
          :T_SERIAL,
          :V_NO,
          :V_DATE,
          :RESPONDING_DATE,
          :T_CODE,
          :R_BRANCH_NO,
          :DESCR1,
          :DR_AMOUNT,
          :CR_AMOUNT,
          SYSDATE,
          :SEND_FLAG,
          :P_FLAG
        );
       commit_form;
  elsif Pflag=1 and OnOf=False Then
        but:=show_alert('NoWay');
        if but=alert_button1 then
      Commit_form;
        else
          raise form_trigger_failure;
        end if;
  ElsIf PFlag=2 and OnOf=True Then
        Commit_form;
        :Send_Flag:=1;
        Transfer_Data
        (
          :BRANCH_NO,
          :AB_FALG,
          :T_DATE,
          :T_SERIAL,
          :V_NO,
          :V_DATE,
          :RESPONDING_DATE,
          :T_CODE,
          :R_BRANCH_NO,
          :DESCR1,
          :DR_AMOUNT,
          :CR_AMOUNT,
          SYSDATE,
          :SEND_FLAG,
```

```
            :P_FLAG
          );
          Commit;
    ElsIf PFlag=2 and OnOf=False Then
          Commit_form;
    ElsIf PFlag=3 and OnOf=True Then
         :Send_Flag:=1;
          Transfer_Data
          (
            :BRANCH_NO,
            :AB_FALG,
            :T_DATE,
            :T_SERIAL,
            :V_NO,
            :V_DATE,
            :RESPONDING_DATE,
            :T_CODE,
            :R_BRANCH_NO,
            :DESCR1,
            :DR_AMOUNT,
            :CR_AMOUNT,
            SYSDATE,
            :SEND_FLAG,
            :P_FLAG
          );
         commit_form;
    ElsIf PFlag=3 and OnOf=False Then
      Commit_form;
    end if;
  else
        but:=show_alert('NoPFlag');
        Raise Form_Trigger_Failure;
        end if;
  clear_form;
  select branch_no into :branch_no from settings;
  select branch_name into :braname1 from branches
  where branch_no=:branch_no;
end;
```

# Form LazyBack

## WHEN-NEW-FORM-INSTANCE

```
Declare
LazyBackTimer Timer;
Five_Secods NUMBER(5) := 5000;
cname varchar2(150);
tit varchar2(200);
Begin
```

```
  LazyBackTimer := CREATE_TIMER('Lazy_timer', Five_Secods, REPEAT);
  Set_Window_Property('MainFWin',Window_State,Maximize);
  Set_Window_Property(Forms_Mdi_Window,Title,'');
  Set_Window_Property(Forms_Mdi_Window,Window_State,Maximize);
End ;
```

WHEN-TIMER-EXPIRED

```
declare
     cnt number;
begin
    select count(*) into cnt from ab_daily_trans where send_flag=0;
    if cnt>0 then
       if isOnline Then
           Lazy_Background;
           Commit;
       end if;
    end if;
 end;
```