DTGen GUI Demonstration

Developed by DMSTEX (http://dmstex.com)

Table of Contents

Introduction:

The exercises in this demonstration are focused on the Graphical User Interface (GUI) "helpers" produced by DTGen. The database schema and data are an extension of the ASOF Demonstration. The "ASOF" demonstration should be reviwed before running these exercises. Serveral concepts introduced in those exercises are not explained here.

With the exception of executing generated scripts, all the functionality in these exercises is available through both command line and graphical user interface (GUI) mode. Issue #23 is an enhancement request to allow executing generated scripts in the GUI. Command line is used in these exercises to expedite progress where needed.

The exercises in this directory are numbered and must be executed in sequential order. The demonstration users must be created with the "create_demo_users.sql" script in the parent directory before the first exercise is run. The demonstration users must be dropped with the "drop_demo_users.sql" script before the "create_demo_users.sql" script can be re-run. These exercises also assume that the default username/password (dtgen/dtgen) is still in use for the generator. Names and passwords are set in the "vars.sql" script and can be modified, if necessary. Also, the DTGen database objects must be installed in the database and the DTGen must be ready to generate code.

Exercise Setup

Command Line:

```
sqlplus system/password @setup
```

Exercise Setup modifies the database. However, the changes are mde in APEX

The setup.sql script creates an APEX workspace and user. These need to be in place before the target application can be created in Exercise #1.

```
Workspace and APEX User DTGEN_DB_DEMO have been setup.

PL/SQL procedure successfully completed.
```

The output above is the result of a successful execution. Note, setup.sql can also be run as sys or APEX_04000. Also, there is a cleanup.sql script that can be used to remove this setup in APEX. Note that identical database schema objects can be dropped and recreated without dropping and recreating APEX objects.

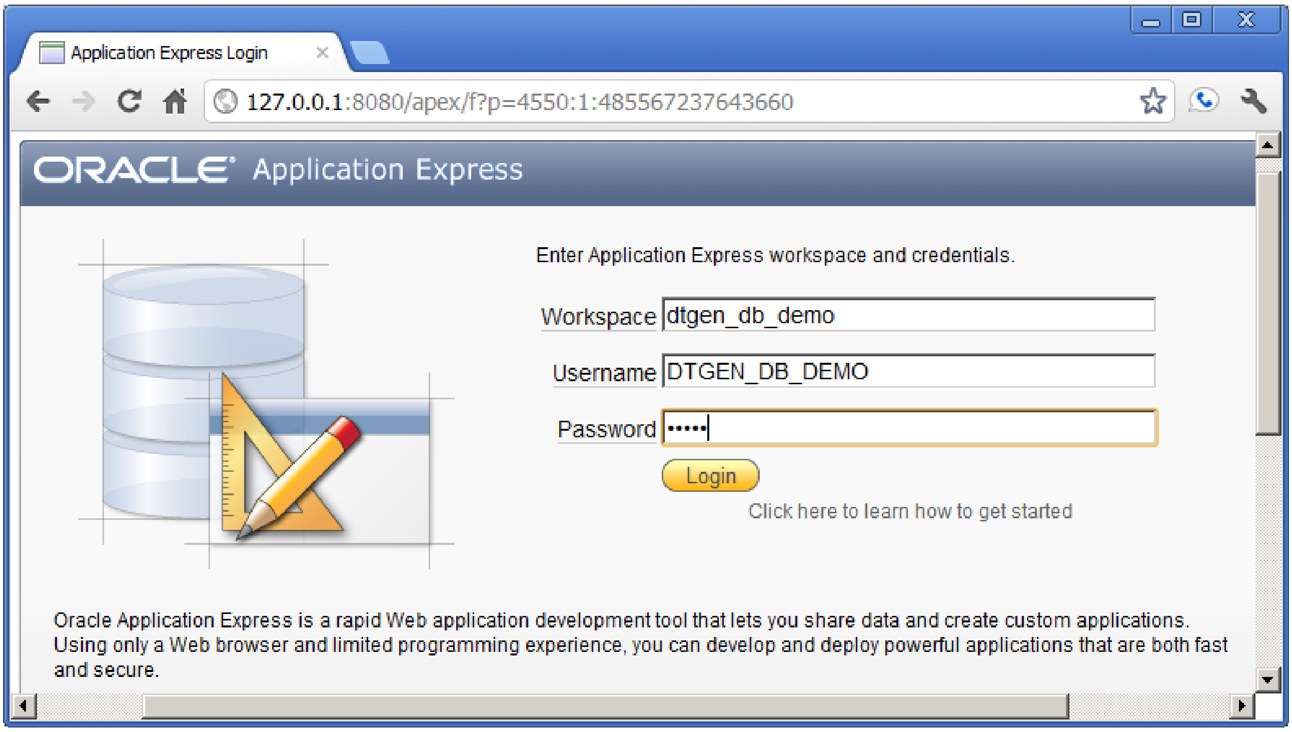
Exercise #1: Default Maintenance Forms

Command Line:

```
APEX Forms
sqlplus /nolog @e1
APEX Forms
```

Exercise #1 modifies the database. The "drop_demo_users.sql" and "create_demo_users.sql" scripts must be used to reset the database before re-running this exercise. The APEX application that is created can be removed or remain as the create_flow.sql scripts will "work around" the existing application.

In this exercise, an empty application will be created in APEX. The database schema and maintenance forms will be generated and loaded into the empty application. Then, navigation lists and breadcrumbs will be updated to integrate the genreated items into the application. The process starts with a login to APEX. The password is "dtgen".



Application Express Login
127.0.0.1:8080/apex/f?p=4550:1:485567237643660
ORACLE Application Express

Enter Application Express workspace and credentials.

Workspace dtgen_db_demo

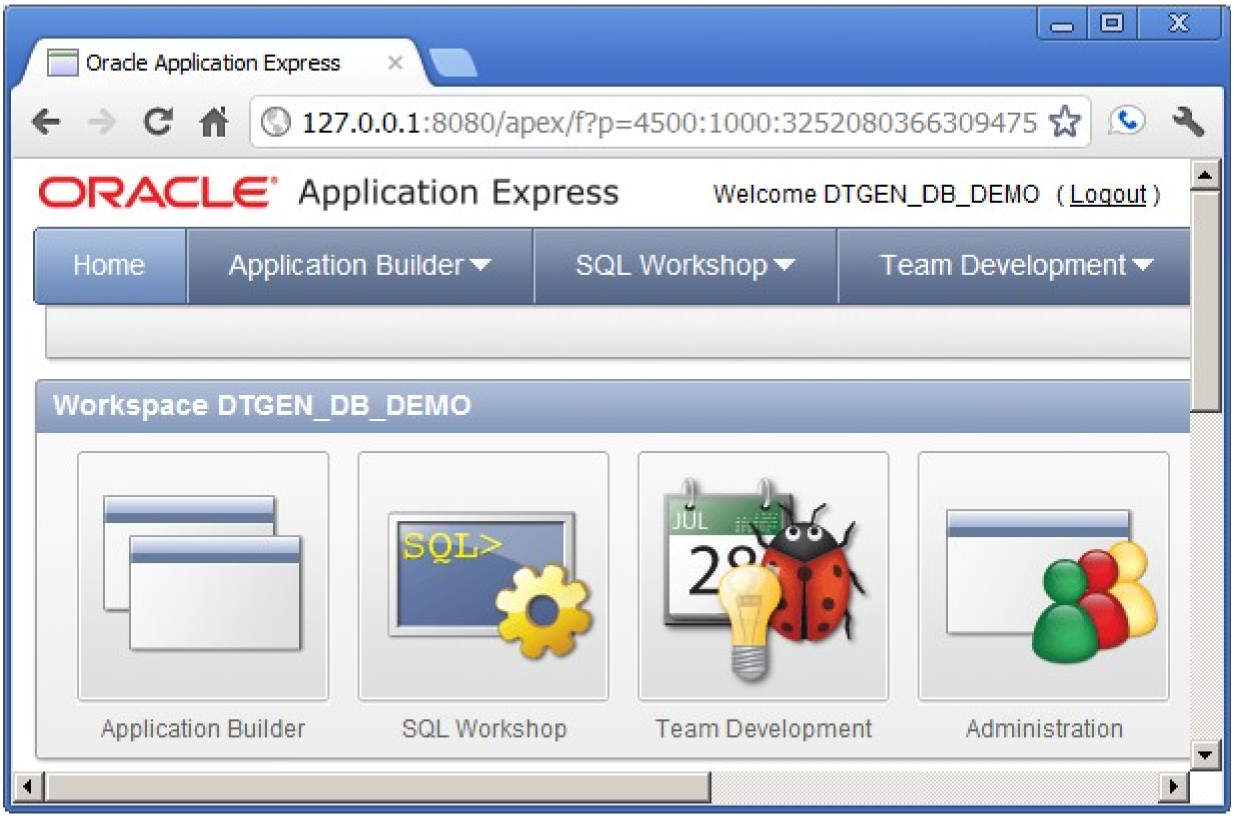
Username DTGEN_DB_DEMO

Password •••••

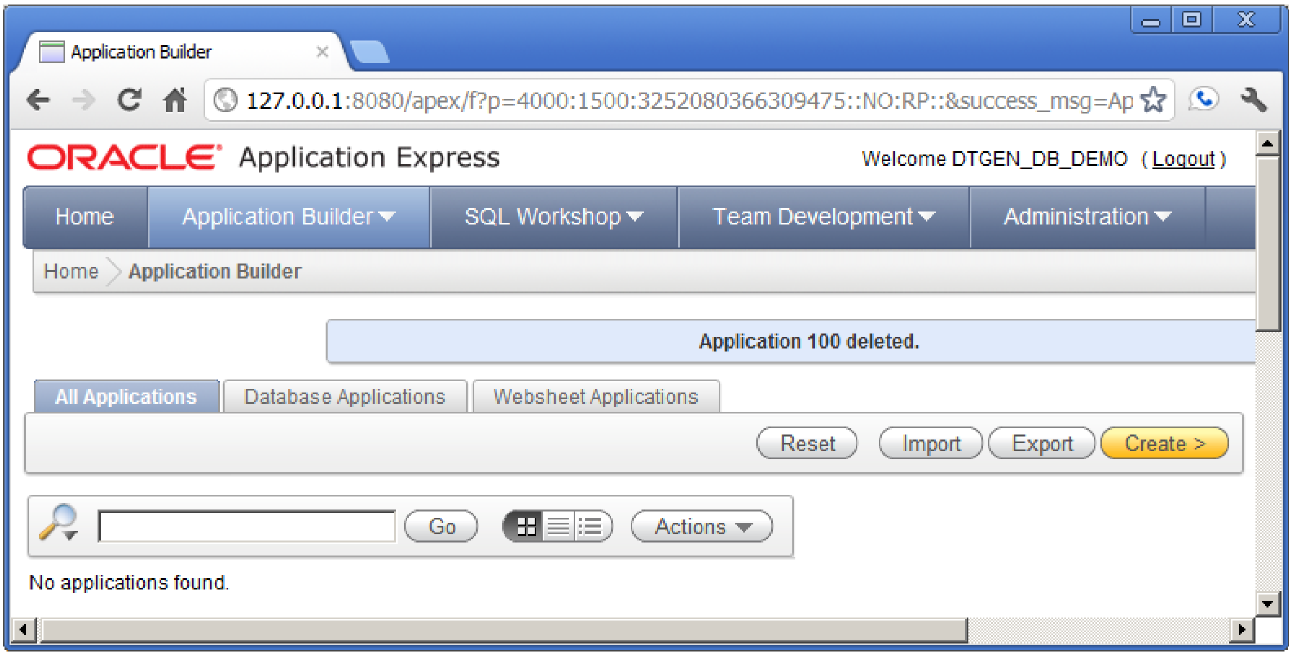
Login
Click here to learn how to get started

Oracle Application Express is a rapid Web application development tool that lets you share data and create custom applications. Using only a Web browser and limited programming experience, you can develop and deploy powerful applications that are both fast and secure.

Click the "login" button to login.



Click the "Application Builder" button.



127.0.0.1:8080/apex/f?p=4000:1500:3252080366309475::NO:RP::&success_msg=Ap

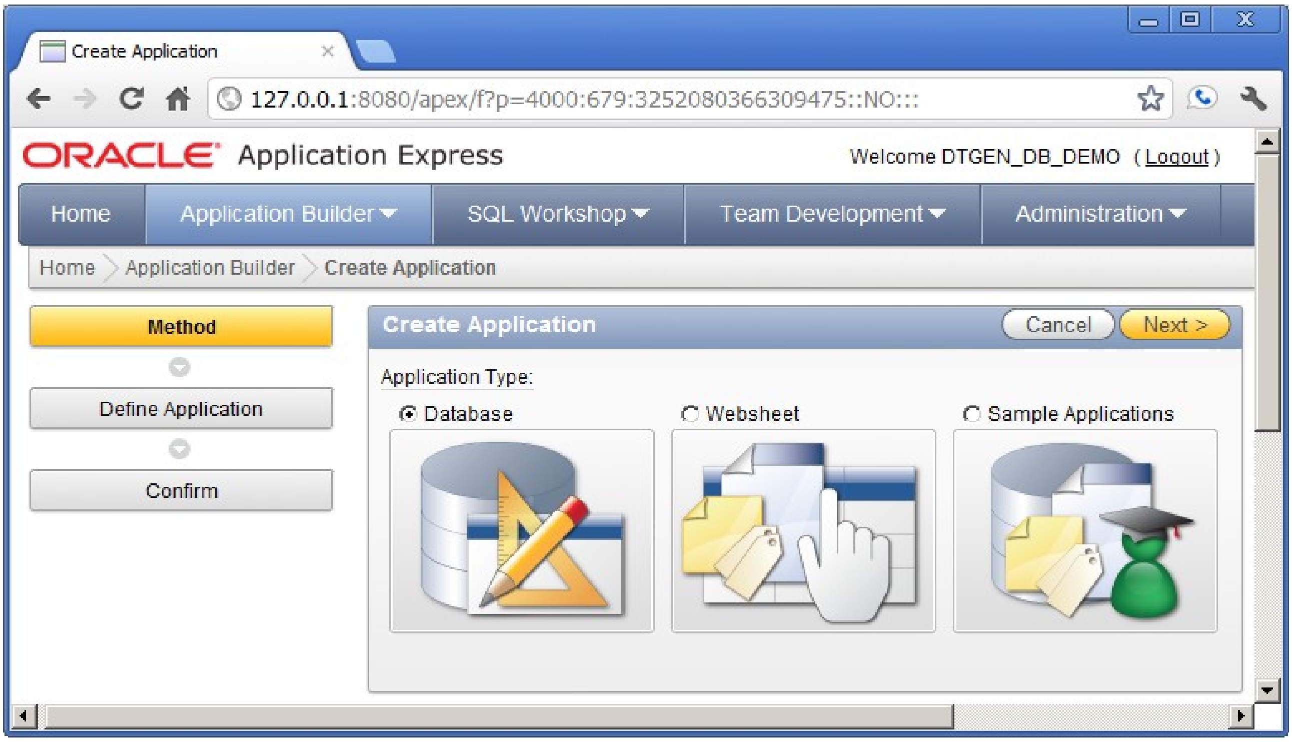
ORACLE Application Express

Home > Application Builder

Application 100 deleted.

No applications found.

Click the "create" button to create a new application.



127.0.0.1:8080/apex/f?p=4000:679:3252080366309475::NO:::

ORACLE Application Express

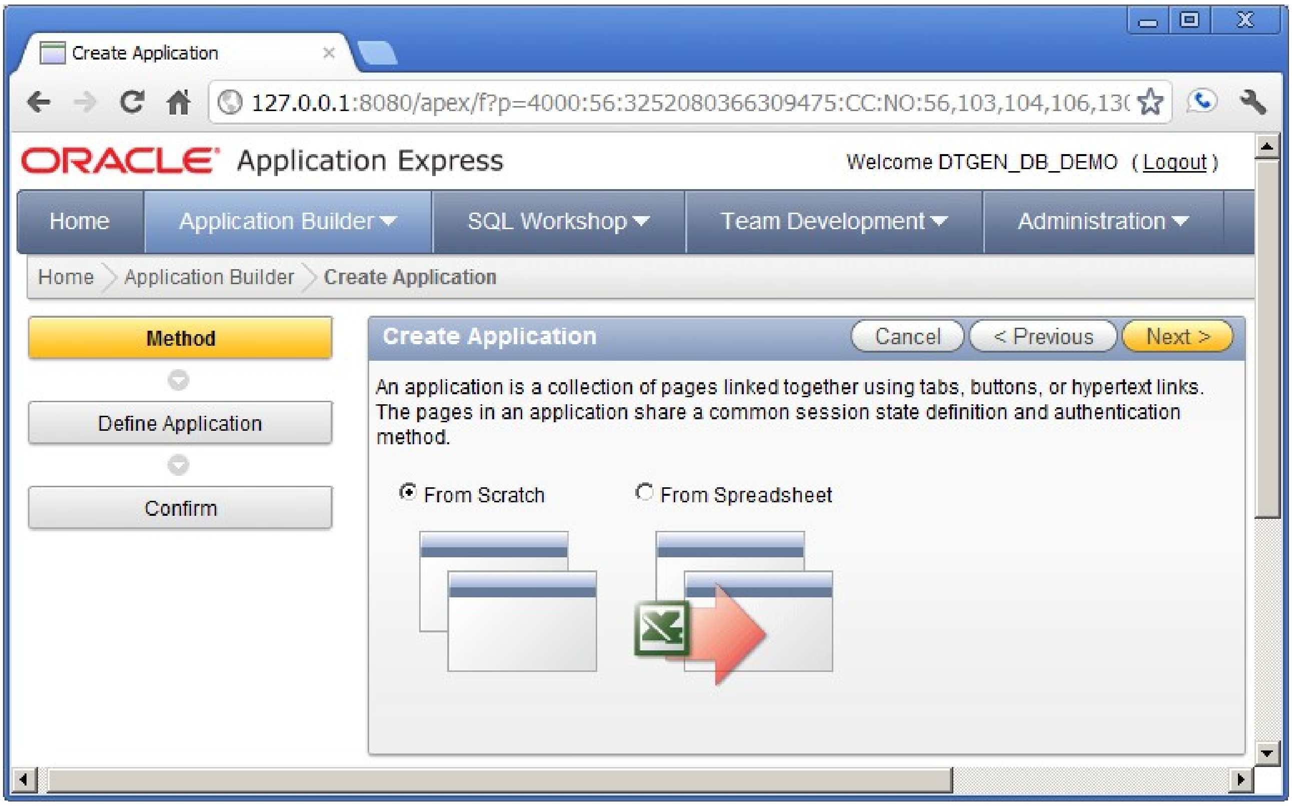
Home > Application Builder > Create Application

Create Application

Application Type:

● Database ○ Websheet ○ Sample Applications

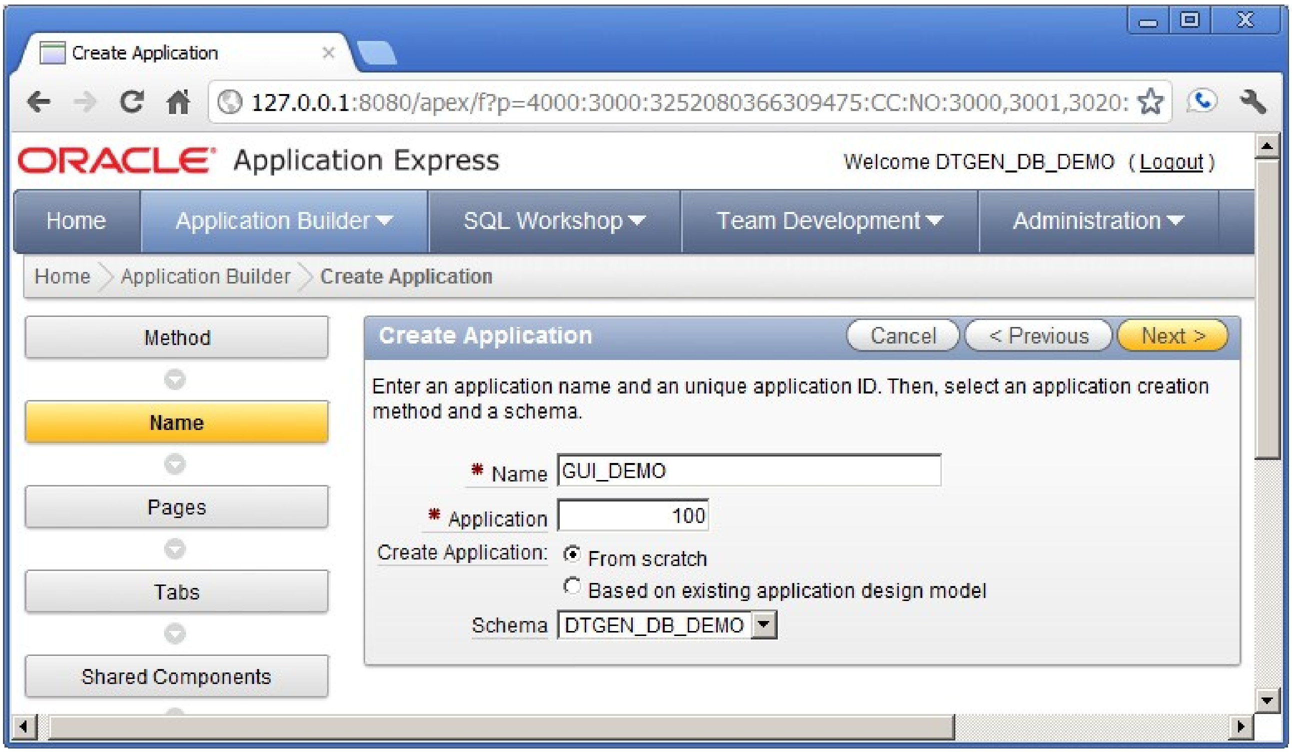
Click the "Database" or the "Next" button.


Create Application
127.0.0.1:8080/apex/f?p=4000:56:3252080366309475:CC:NO:56,103,104,106,13(
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Create Application
Method
Define Application
Confirm
Create Application
Cancel
< Previous
Next >

An application is a collection of pages linked together using tabs, buttons, or hypertext links. The pages in an application share a common session state definition and authentication method.

● From Scratch ○ From Spreadsheet

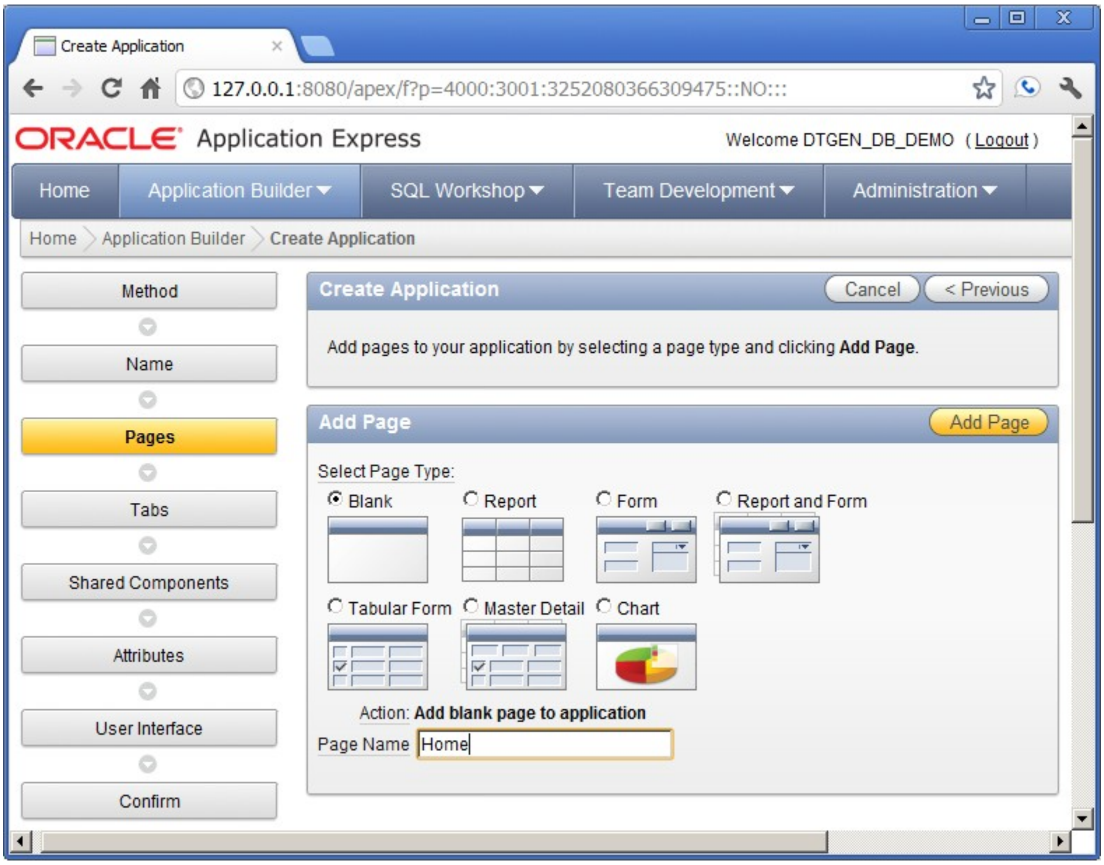
Click the "From Scratch" or the "Next" button.


Create Application
127.0.0.1:8080/apex/f?p=4000:3000:3252080366309475:CC:NO:3000,3001,3020:
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Create Application
Method
Name
Pages
Tabs
Shared Components
Create Application
Cancel
< Previous
Next >

Enter an application name and an unique application ID. Then, select an application creation method and a schema.

* Name	GUI_DEMO
* Application	100
Create Application:	● From scratch ○ Based on existing application design model
Schema	DTGEN_DB_DEMO

Update the values and click the "Next" button.



Add pages to your application by selecting a page type and clicking **Add Page**.

Select Page Type:

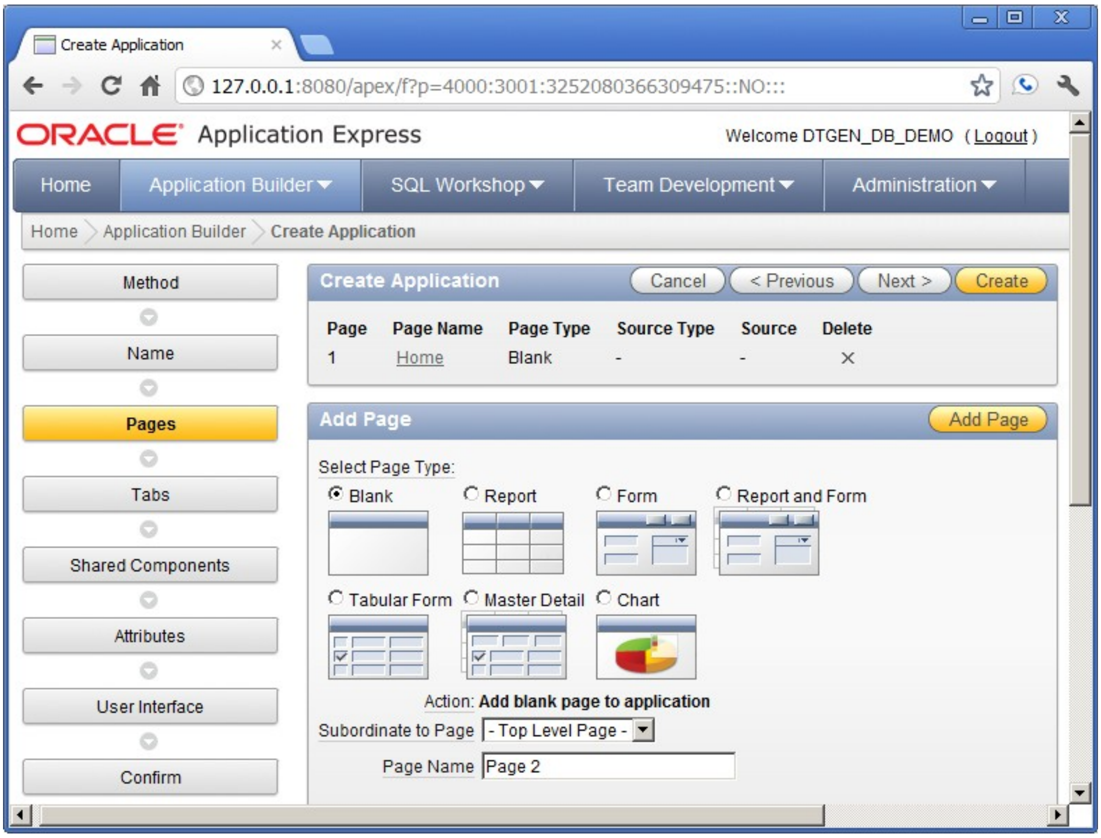
● Blank ○ Report ○ Form ○ Report and Form

○ Tabular Form ○ Master Detail ○ Chart

Action: **Add blank page to application**

Page Name: Home

Update the values and click "Add Page".



Page	Page Name	Page Type	Source Type	Source	Delete
1	Home	Blank	-	-	×

Select Page Type:

● Blank ○ Report ○ Form ○ Report and Form

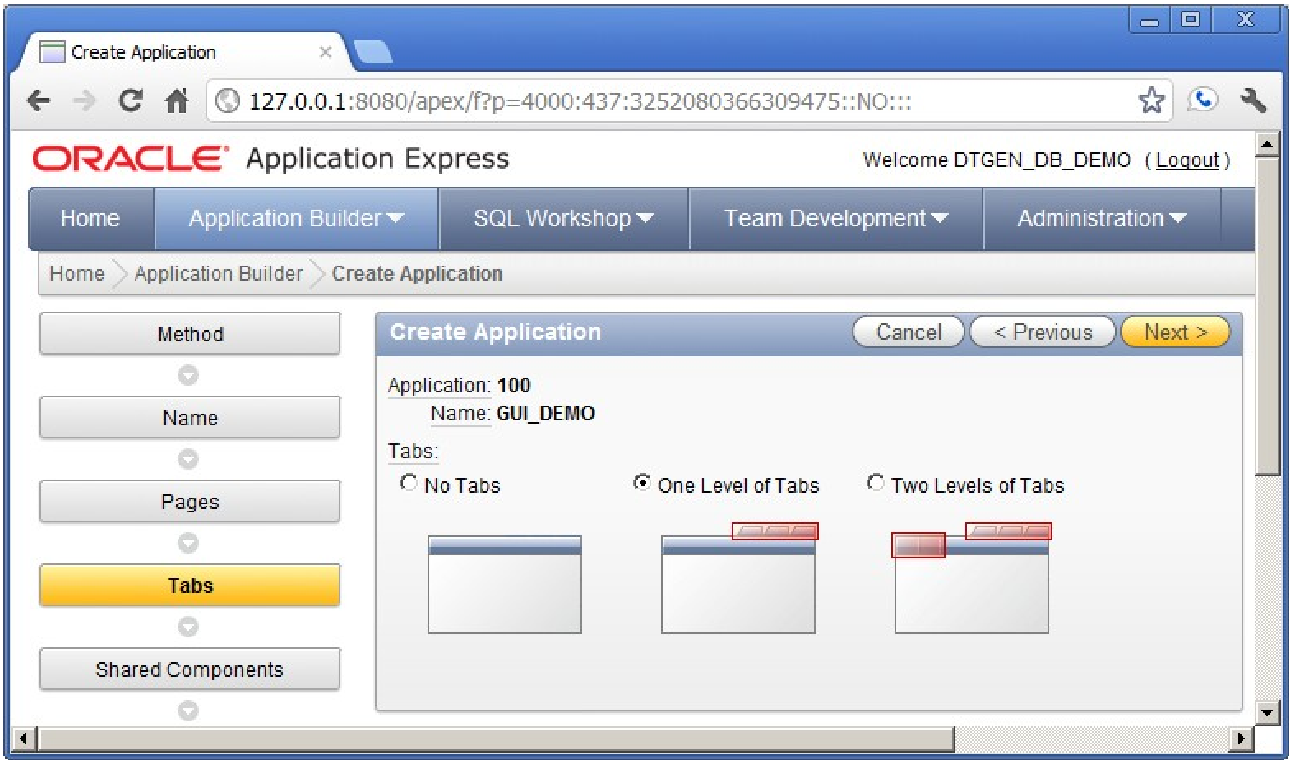
○ Tabular Form ○ Master Detail ○ Chart

Action: **Add blank page to application**

Subordinate to Page: - Top Level Page -

Page Name: Page 2

Confirm the values and click the "Next" button


Create Application
127.0.0.1:8080/apex/f?p=4000:437:3252080366309475::NO:::
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Create Application
Method
Name
Pages
Tabs
Shared Components
Cancel
< Previous
Next >

Create Application

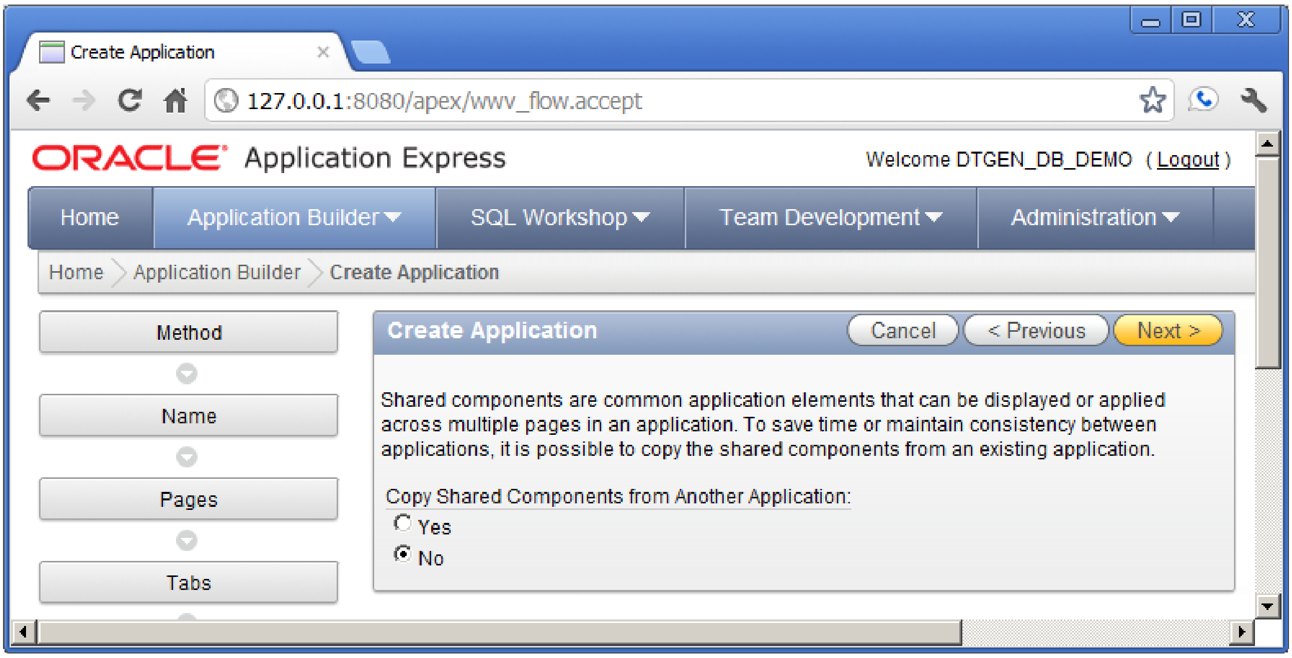
Application: 100

Name: GUI_DEMO

Tabs:

○ No Tabs ● One Level of Tabs ○ Two Levels of Tabs

Confirm the values and click the "Next" button


Create Application
127.0.0.1:8080/apex/wwv_flow.accept
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Create Application
Method
Name
Pages
Tabs
Cancel
< Previous
Next >

Create Application

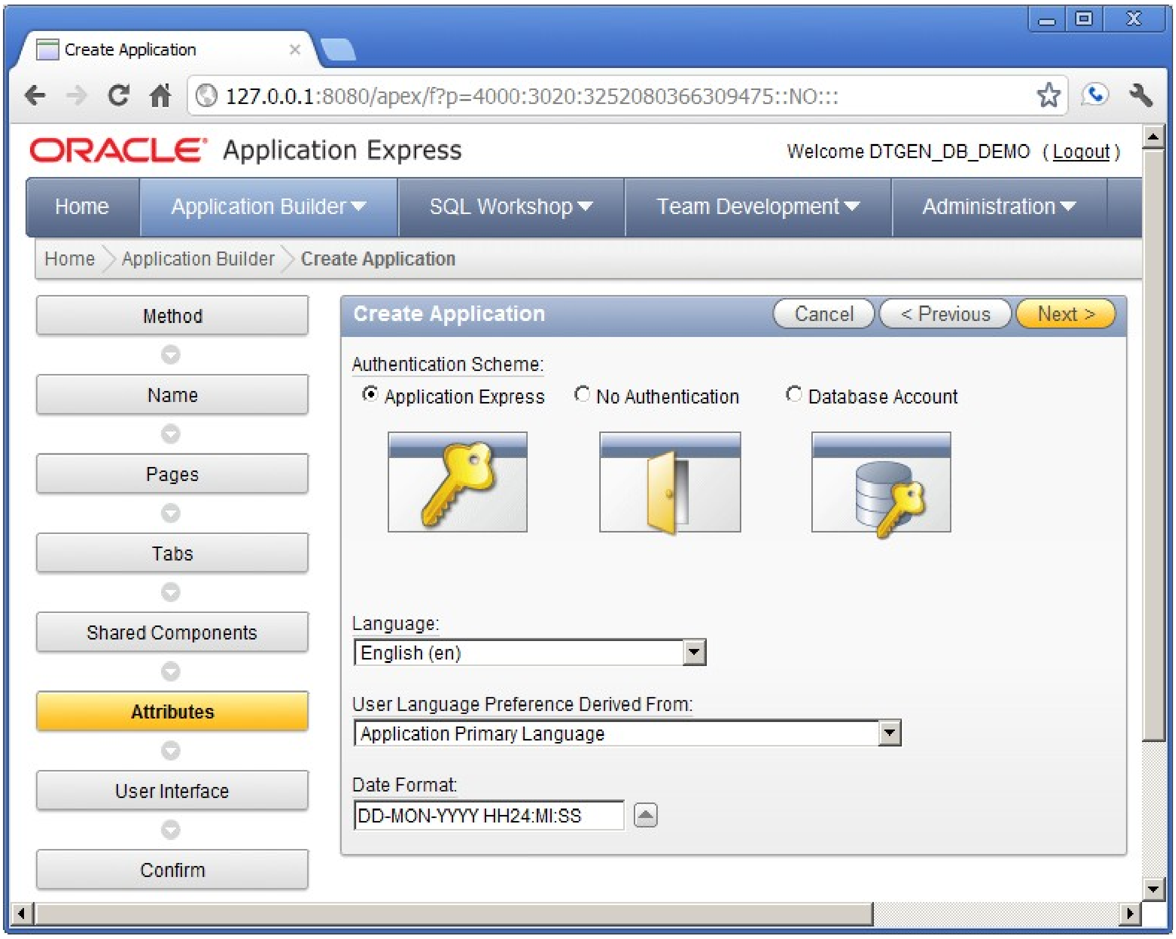
Shared components are common application elements that can be displayed or applied across multiple pages in an application. To save time or maintain consistency between applications, it is possible to copy the shared components from an existing application.

Copy Shared Components from Another Application:

○ Yes

● No

Confirm the values and click the "Next" button.



Create Application

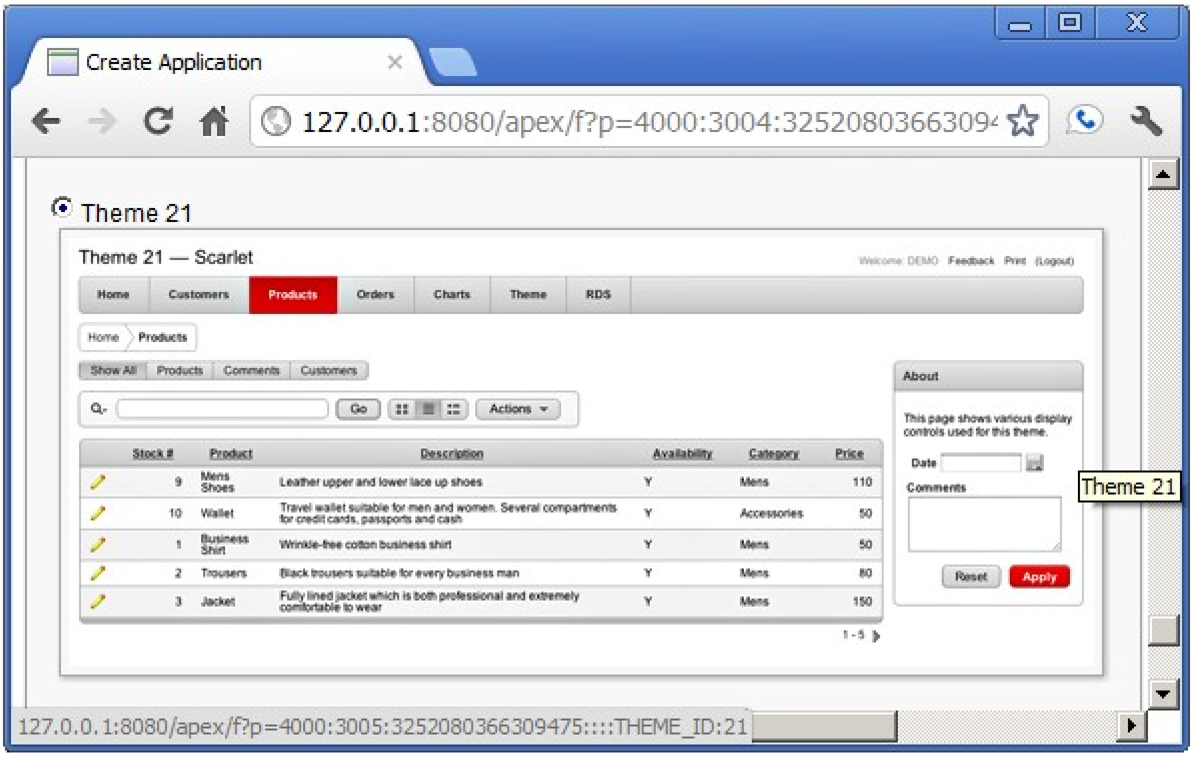
Authentication Scheme:
● Application Express ○ No Authentication ○ Database Account

Language: English (en)

User Language Preference Derived From: Application Primary Language

Date Format: DD-MON-YYYY HH24:MI:SS

Enter the values and click "Next"

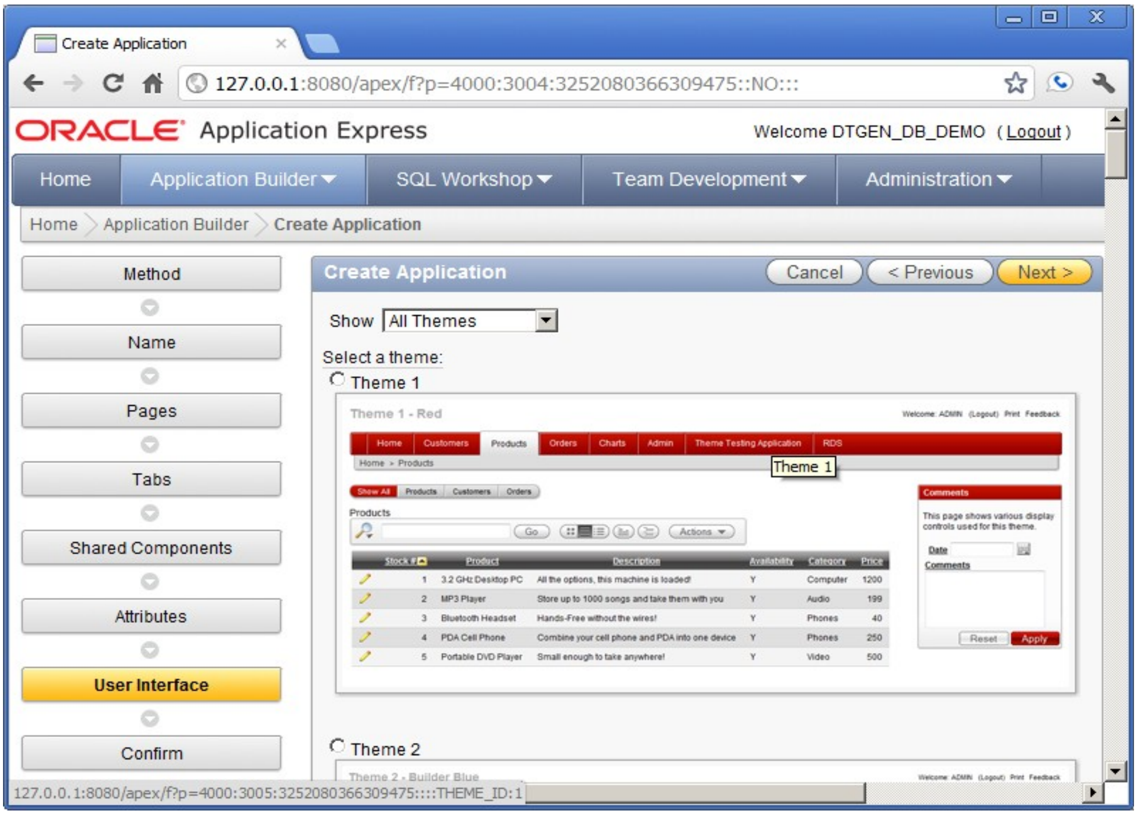

Create Application
127.0.0.1:8080/apex/f?p=4000:3004:3252080366309
Theme 21 — Scarlet
Welcome: DEMO Feedback Print (Logout)
Home
Customers
Products
Orders
Charts
Theme
RDS
Home > Products
Show All
Products
Comments
Customers
Go
Actions
About
This page shows various display controls used for this theme.
Date
Comments
Reset
Apply
Theme 21
127.0.0.1:8080/apex/f?p=4000:3005:3252080366309475::::THEME_ID:21

● Theme 21

	Stock #	Product	Description	Availability	Category	Price
	9	Mens Shoes	Leather upper and lower lace up shoes	Y	Mens	110
	10	Wallet	Travel wallet suitable for men and women. Several compartments for credit cards, passports and cash	Y	Accessories	50
	1	Business Shirt	Wrinkle-free cotton business shirt	Y	Mens	50
	2	Trousers	Black trousers suitable for every business man	Y	Mens	80
	3	Jacket	Fully lined jacket which is both professional and extremely comfortable to wear	Y	Mens	150

1 - 5

Confirm that Theme 21 is selected.



Create Application
127.0.0.1:8080/apex/f?p=4000:3004:3252080366309475::NO:::
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Create Application
Method
Name
Pages
Tabs
Shared Components
Attributes
User Interface
Confirm
Cancel
< Previous
Next >
Theme 1
127.0.0.1:8080/apex/f?p=4000:3005:3252080366309475::::THEME_ID:1

Create Application

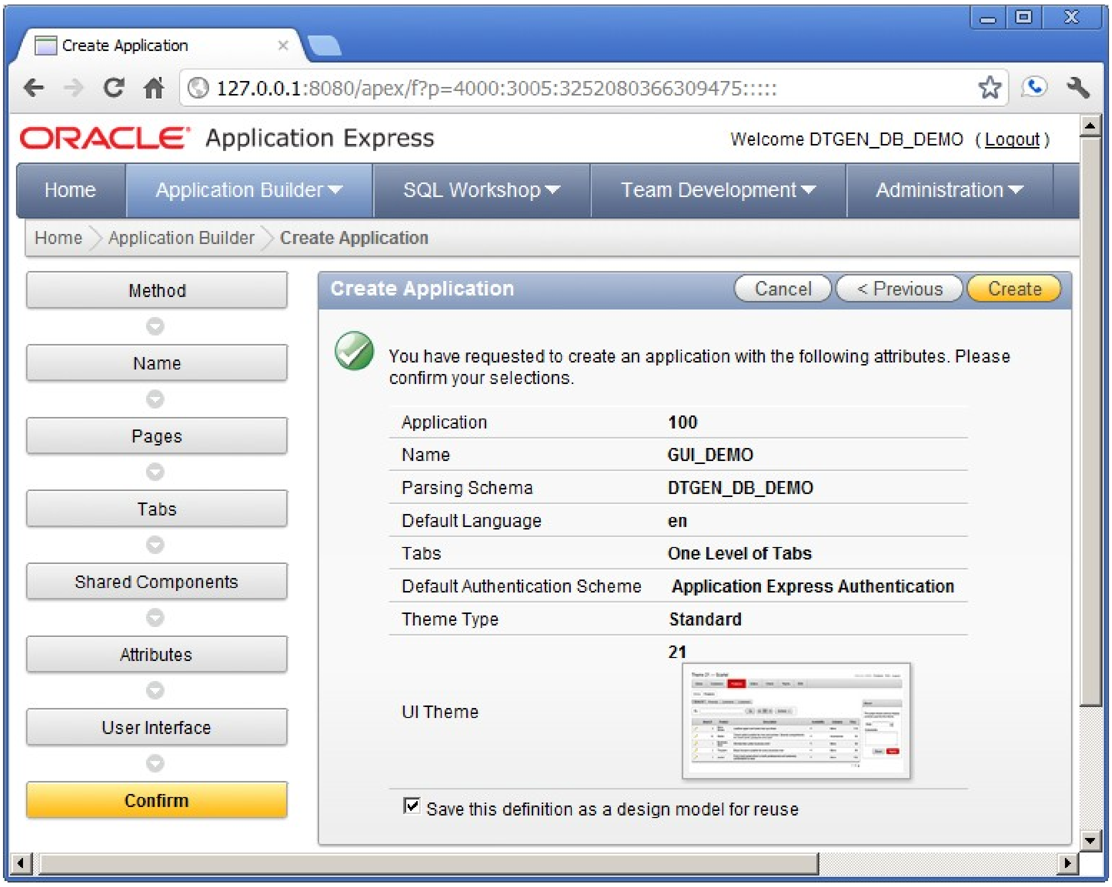
Show All Themes

Select a theme:

○ Theme 1

○ Theme 2

The screen above is the same as the previous screen. Confirm and click the "Next" button.



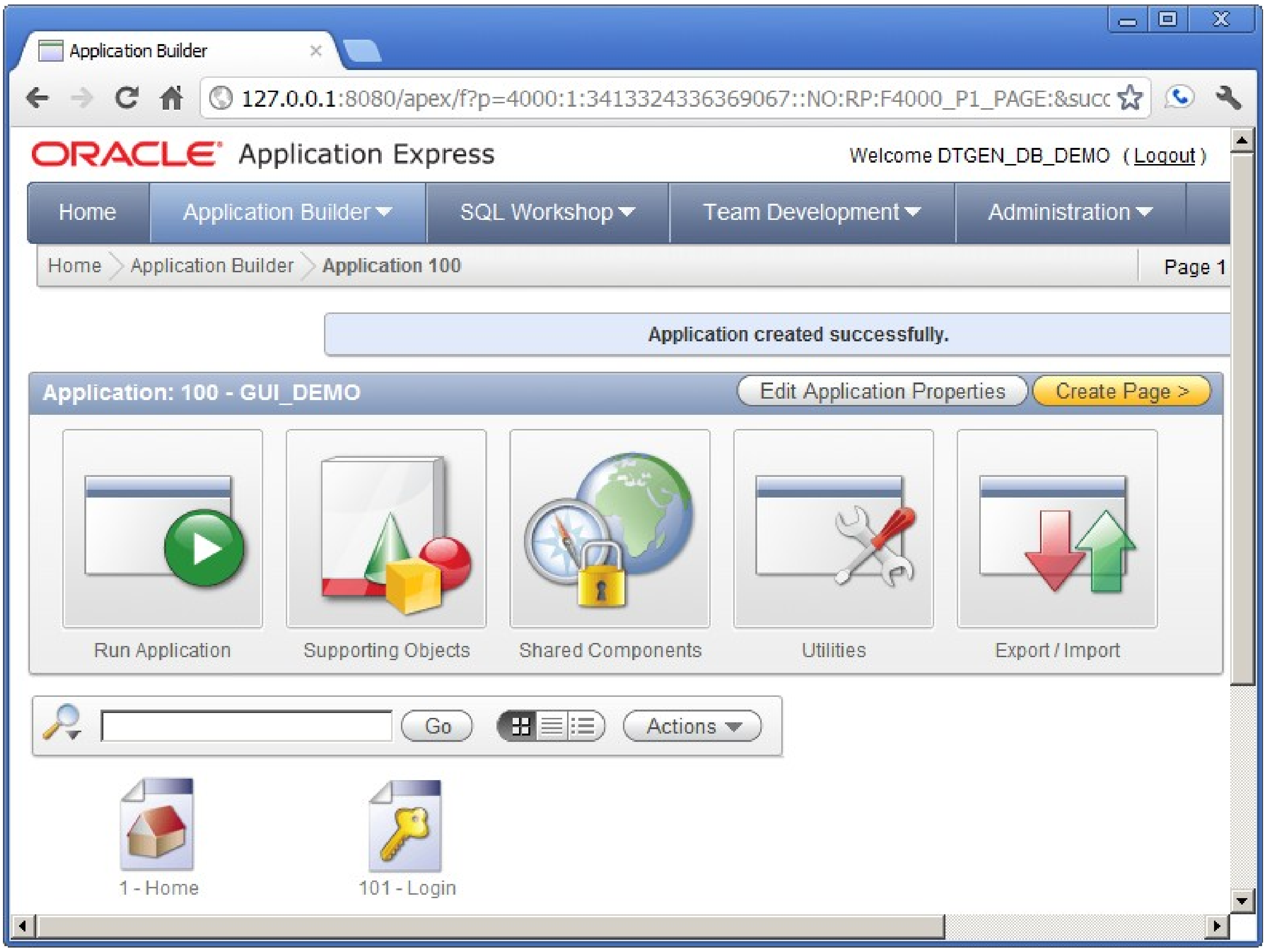
Create Application

You have requested to create an application with the following attributes. Please confirm your selections.

Application	100
Name	GUI_DEMO
Parsing Schema	DTGEN_DB_DEMO
Default Language	en
Tabs	One Level of Tabs
Default Authentication Scheme	Application Express Authentication
Theme Type	Standard
UI Theme	21

☑ Save this definition as a design model for reuse

Enter the values and click the "Create" button.



Application Builder
127.0.0.1:8080/apex/f?p=4000:1:3413324336369067::NO:RP:F4000_P1_PAGE:&succ
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Application 100
Page 1

Application created successfully.

Application: 100 - GUI_DEMO
Edit Application Properties
Create Page >
Run Application
Supporting Objects
Shared Components
Utilities
Export / Import
Go
Actions
1 - Home
101 - Login

The new application has been created and is ready to receive the generated items from DTGen. The following is the results from executing the e1.sql script.

```
Login to dtgen
Connected.
Remove old DEMO4 Schema from DTGEN
create a DEMO4 Schema in DTGEN
Generate DEMO4 Application
Capture SQL Scripts
```

Similar to the other demonstrations, the output shown above is from the e1.LST log file. The details of modifying schema in DTGen and generating will be covered in following exercises. Below is the output from the install.LST log file that follows the SQL script capture in e1.sql.

```
Login to dtgen_db_demo
Connected.

FILE_NAME
-----------------
 -) create_glob

FILE_NAME
----------------
 -) create_ods

TABLE_NAME
--------------
***  dept  ***

TABLE_NAME
-------------
```

```
*** emp ***

FILE_NAME
------------------
 -) create_integ

TABLE_NAME
--------------
***  dept  ***

TABLE_NAME
-------------
***  emp  ***

FILE_NAME
-----------------
 -) create_oltp

TABLE_NAME
--------------
***  dept  ***

TABLE_NAME
-------------
***  emp  ***

FILE_NAME
---------------
 -) create_aa

TABLE_NAME
--------------
***  dept  ***

TABLE_NAME
-------------
***  emp  ***

FILE_NAME
-----------------
 -) create_mods
```

The portion of the e1.sql script that generated the output above is similar to Exercise #1 in the other demonsrations. However, the following portion of the install.LST log file shows the APEX GUI items that were generated by DTGen being installed.

```
FILE_NAME
-----------------
 -) create_flow
   APPLICATION 100 - GUI_DEMO
  Set Credentials...
  Check Compatibility...
API Last Extended:20100513
Your Current Version:20100513
This import is compatible with version: 20100513
COMPATIBLE (You should be able to run this import without issues.)

  Set Application ID...
  ...application processes
  ...Create Flow Process "SET_USR"
  ...Shared Lists of values
  ...Create LOV "JOB NAME"
  ...Remove page 1000
  ...Create page 1000: Maintenance Menu
  ...Navigation Tabs (for Maintenance Menu)
  Adding Maintenance Menu Tab to UTIL_TS ...
  ...Navigation Lists
  ...Create LIST " Maintenance Menu"
  ...Create LIST ENTRY "Dept Maint"
  ...Create LIST ENTRY "Emp Maint"
  ... Add LIST " Maintenance Menu" to page 1000
  ...Remove page 1200
  ...Create page 1200: Utility Log Report
  ...Create UTIL_LOG Report Plug
  ...Create Page Processing
  ...Navigation Tabs (for Utility Log Report)
  Adding Utility Log Report Tab to UTIL_TS ...
  ...Remove page 1400
  ...Create page 1400: OMNI Reports Menu
  ...Navigation Tabs (for OMNI Reports Menu)
  Adding T_OMNI Tab to UTIL_TS ...
  ...Navigation Lists
  ...Create LIST " OMNI Reports Menu"
  ...Create LIST ENTRY "Dept OMNI"
  ...Create LIST ENTRY "Emp OMNI"
  ... Add LIST " OMNI Reports Menu" to page 1400
  ...Remove page 1600
  ...Create page 1600: ASOF Reports Menu
  ...Navigation Tabs (for ASOF Reports Menu)
  Adding T_ASOF Tab to UTIL_TS ...
  ...Navigation Lists
  ...Create LIST " ASOF Reports Menu"
  ...Create LIST ENTRY "Dept ASOF"
  ...Create LIST ENTRY "Emp ASOF"
  ... Add LIST " ASOF Reports Menu" to page 1600
  ...Create LIST "Utility Menu"
  ...Create LIST ENTRY "Maintenance Menu"
  ...Create LIST ENTRY "Utility Log Report"
  ...Create LIST ENTRY "OMNI Reports Menu"
  ...Create LIST ENTRY "ASOF Reports Menu"
Adding Breadcrumbs
  ...Create BREADCRUMB ENTRY "Maintenance Menu"
  ...Create BREADCRUMB ENTRY "Dept Maint"
  ...Create BREADCRUMB ENTRY "Dept Form"
  ...Create BREADCRUMB ENTRY "Emp Maint"
  ...Create BREADCRUMB ENTRY "Emp Form"
  ...Create BREADCRUMB ENTRY "Utility Log Report"
  ...Create BREADCRUMB ENTRY "OMNI Reports Menu"
  ...Create BREADCRUMB ENTRY "Dept OMNI"
  ...Create BREADCRUMB ENTRY "Emp OMNI"
  ...Create BREADCRUMB ENTRY "ASOF Reports Menu"
  ...Create BREADCRUMB ENTRY "Dept ASOF"
  ...Create BREADCRUMB ENTRY "Emp ASOF"
```

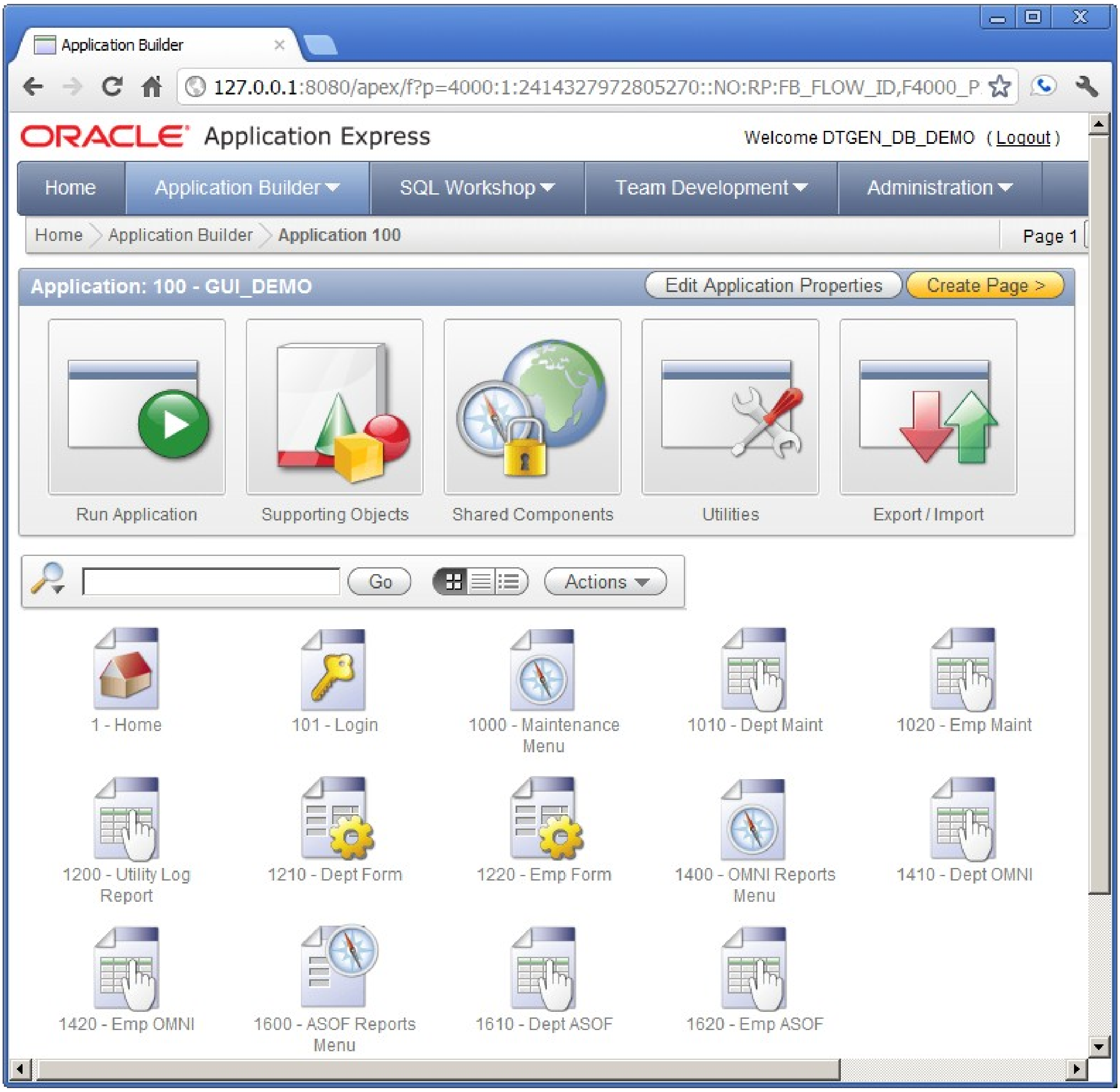
All of the output above is the result of GUI navigation items being installed into APEX. The output from the DEPT and EMP data maintenance forms is below.

```
TABLE_NAME
--------------
***  dept  ***
dept maint1, 10
  ...Create Shared LOV "DEPT_LOV"
  ...Remove page 1010
  ...Create page 1010: Dept Maint
  ...Create Main Grid Edit
  ...Create Filter Criteria
  ...Create Page Processing
  Adding Dept Maint Tab to _MAINT_TS ...
dept maint2, 10
  ...Create ALL Report
  ...Create History Report
  ...Create POP_AUDIT Report
dept form, 10
  ...Remove page 1210
  ...Create page 1210: Dept Form
  ...Create DML Form
  ...Create Page Processing
  Adding Dept Form Tab to _FORM_TS ...
dept omni, 10
  ...Remove page 1410
  ...Create page 1410: Dept OMNI
  ...Create Interactive Report
  Adding Dept OMNI Tab to _OMNI_TS ...
dept asof, 10
  ...Remove page 1610
  ...Create page 1610: Dept ASOF
  ...Create Interactive Report
  Adding Dept ASOF Tab to _ASOF_TS ...
SCHEMA DTGEN_DB_DEMO - User Interface Defaults, Table Defaults
```

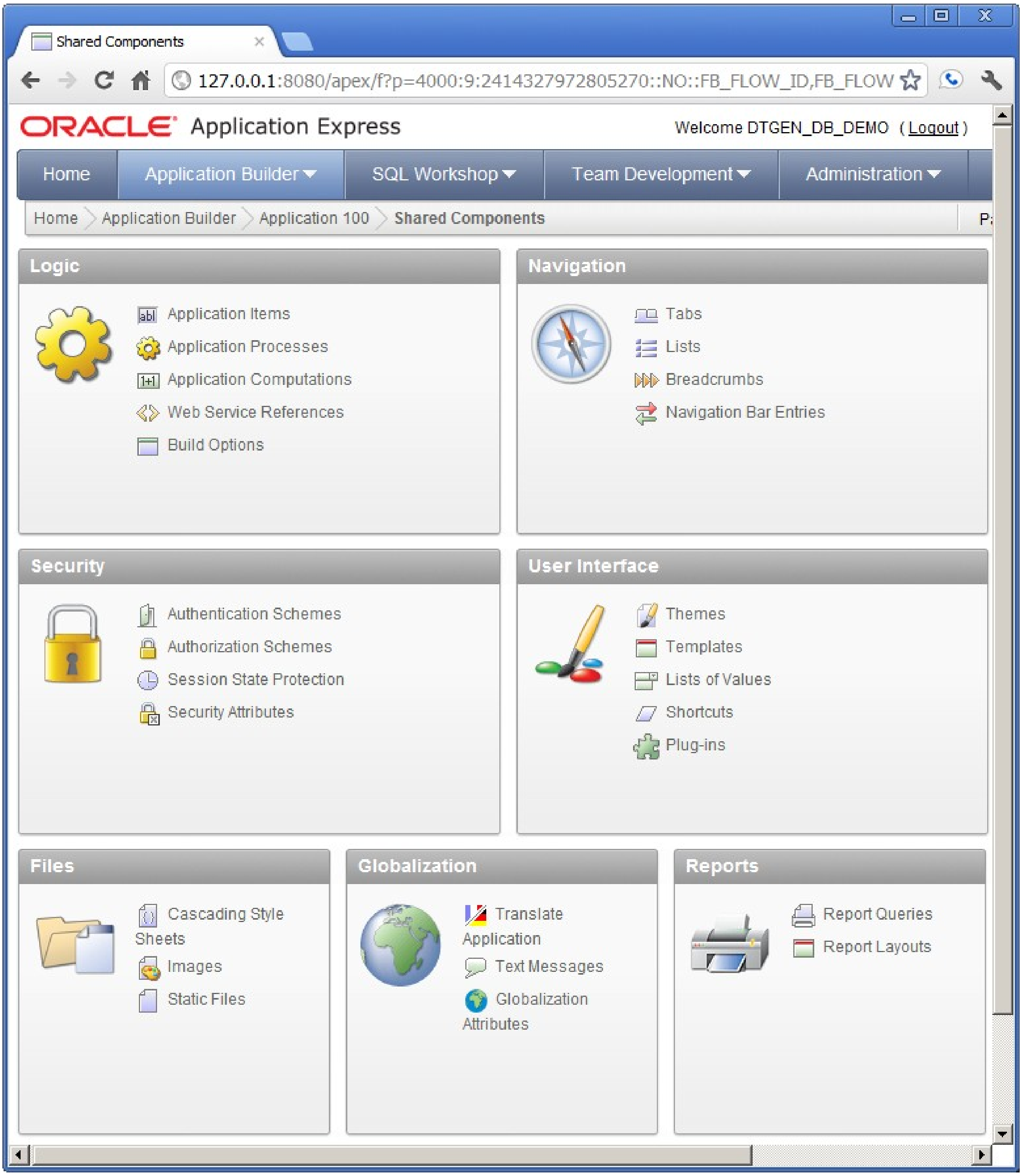
```
TABLE_NAME
-------------
***  emp  ***
emp maint1, 20
  ...Create Shared LOV "EMP_LOV"
  ...Create Shared LOV "MGR_EMP_LOV"
  ...Remove page 1020
  ...Create page 1020: Emp Maint
  ...Create Main Grid Edit
  ...Create Filter Criteria
  ...Create Page Processing
  Adding Emp Maint Tab to _MAINT_TS ...
emp maint2, 20
  ...Create ALL Report
  ...Create History Report
  ...Create POP_AUDIT Report
emp form, 20
  ...Remove page 1220
  ...Create page 1220: Emp Form
  ...Create DML Form
  ...Create Page Processing
  Adding Emp Form Tab to _FORM_TS ...
emp omni, 20
  ...Remove page 1420
  ...Create page 1420: Emp OMNI
  ...Create Interactive Report
  Adding Emp OMNI Tab to _OMNI_TS ...
emp asof, 20
  ...Remove page 1620
  ...Create page 1620: Emp ASOF
  ...Create Interactive Report
  Adding Emp ASOF Tab to _ASOF_TS ...
SCHEMA DTGEN_DB_DEMO - User Interface Defaults, Table Defaults
 - User Interface Defaults, Attribute Dictionary
...done

Loading data into database
```

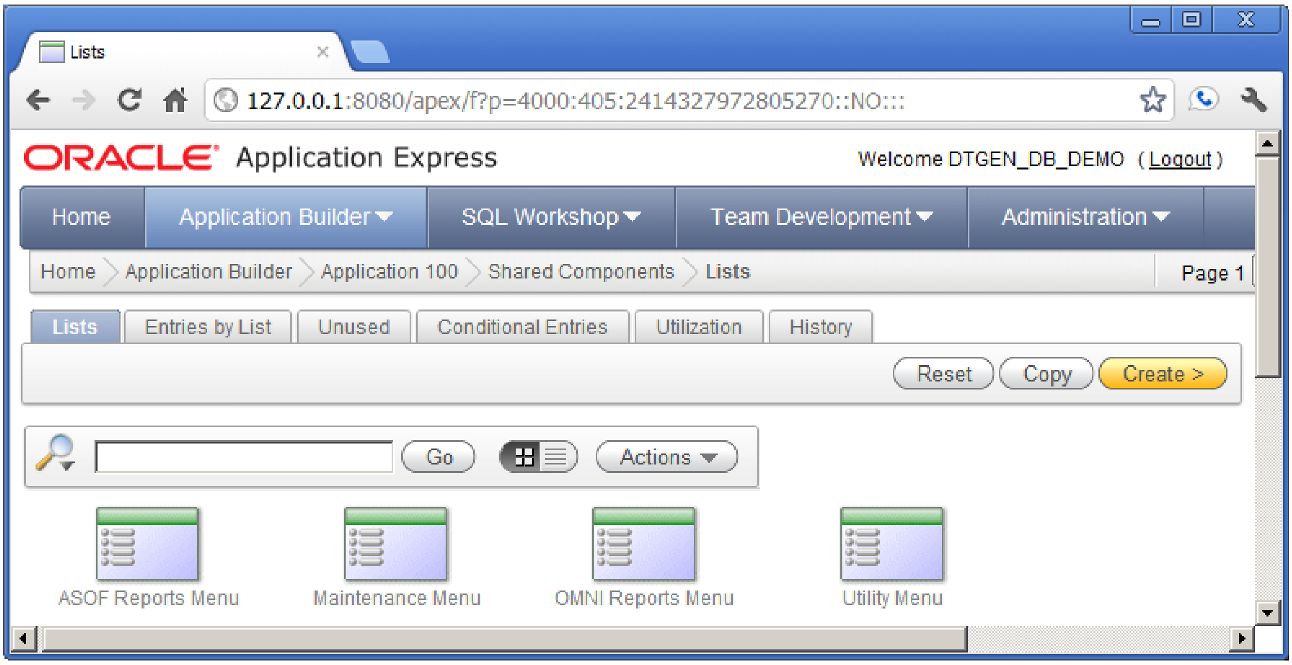
At the end of the output above is the indication that the schema data has been loaded. This is the same data used in the ASOF demonstration. The remaining effort is to "connect" the generated navigation items with the empty application in APEX. This effort starts with a refresh of the application builder page (or login as DTGEN_DB_DEMO and navigate to the GUI_DEMO application page if not logged into APEX).



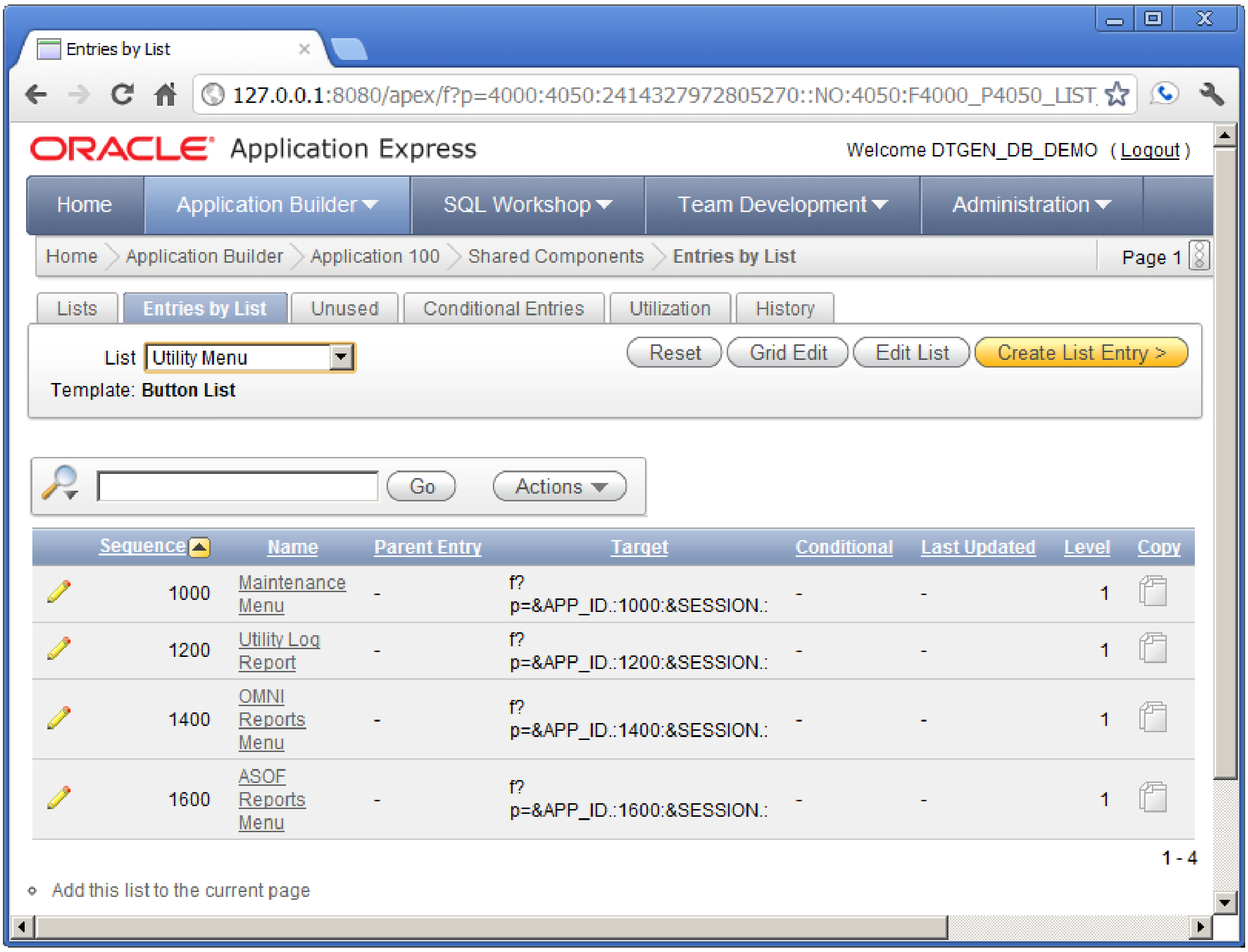
Click the "Shared Components" button



Click the "Lists" link in the "Navigation" pane



Click the "Utility Menu" icon.


Entries by List
127.0.0.1:8080/apex/f?p=4000:4050:2414327972805270::NO:4050:F4000_P4050_LIST
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Application 100 > Shared Components > Entries by List
Page 1
Lists
Entries by List
Unused
Conditional Entries
Utilization
History
Reset
Grid Edit
Edit List
Create List Entry >
Go
Actions

List: Utility Menu

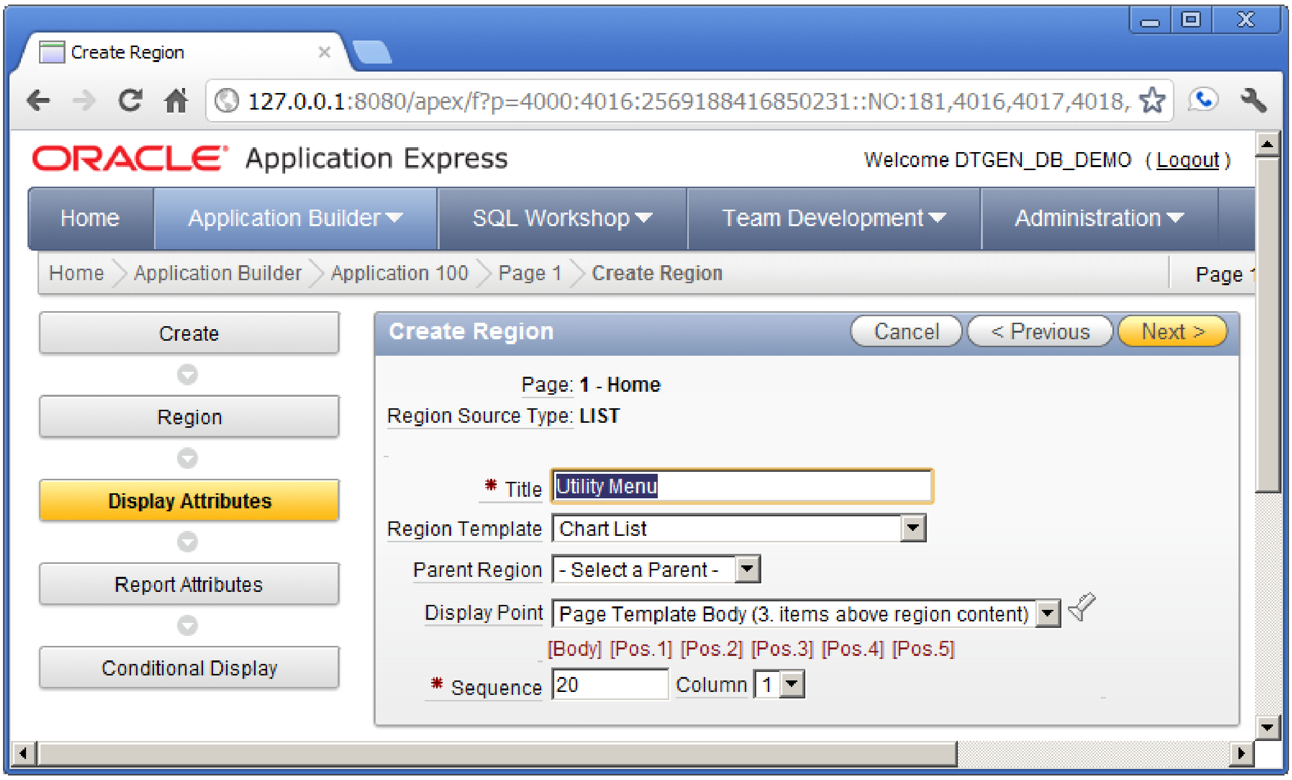
Template: **Button List**

	Sequence	Name	Parent Entry	Target	Conditional	Last Updated	Level	Copy
✎	1000	Maintenance Menu	-	f?p=&APP_ID.:1000:&SESSION.:	-	-	1	
✎	1200	Utility Log Report	-	f?p=&APP_ID.:1200:&SESSION.:	-	-	1	
✎	1400	OMNI Reports Menu	-	f?p=&APP_ID.:1400:&SESSION.:	-	-	1	
✎	1600	ASOF Reports Menu	-	f?p=&APP_ID.:1600:&SESSION.:	-	-	1	

1 - 4

- Add this list to the current page

Confirm the "Page 1" is in the upper right and click the "Add this list to the current page" link.

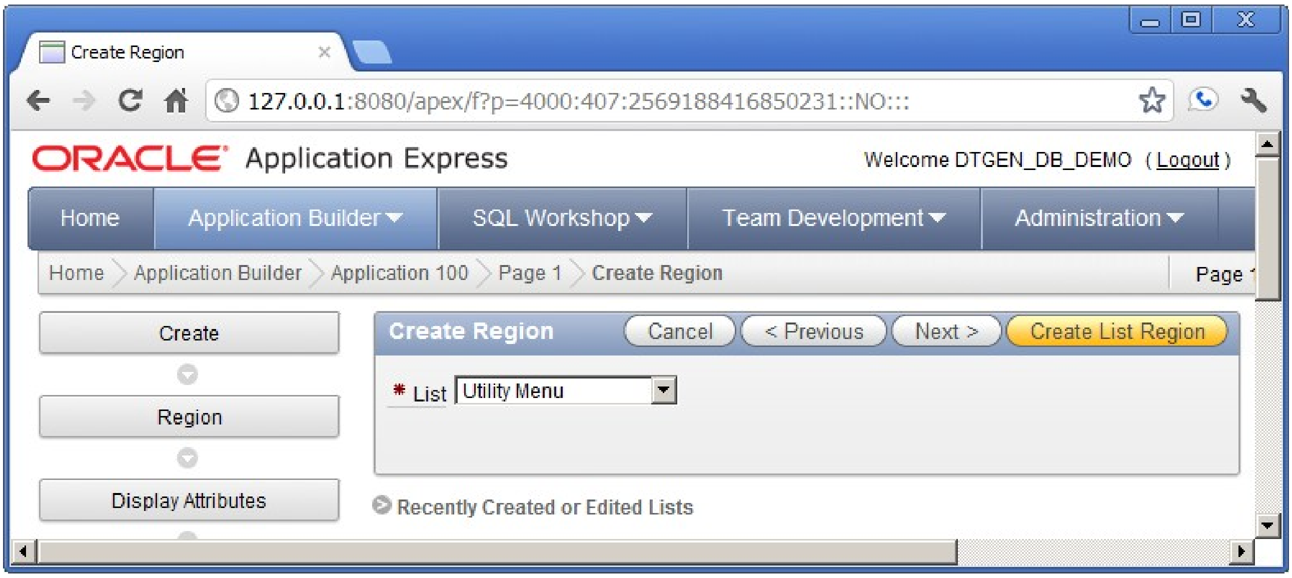

Create Region
127.0.0.1:8080/apex/f?p=4000:4016:2569188416850231::NO:181,4016,4017,4018,
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Application 100 > Page 1 > Create Region
Page 1
Create
Region
Display Attributes
Report Attributes
Conditional Display
Create Region
Cancel
< Previous
Next >

Page: **1 - Home**

Region Source Type: **LIST**

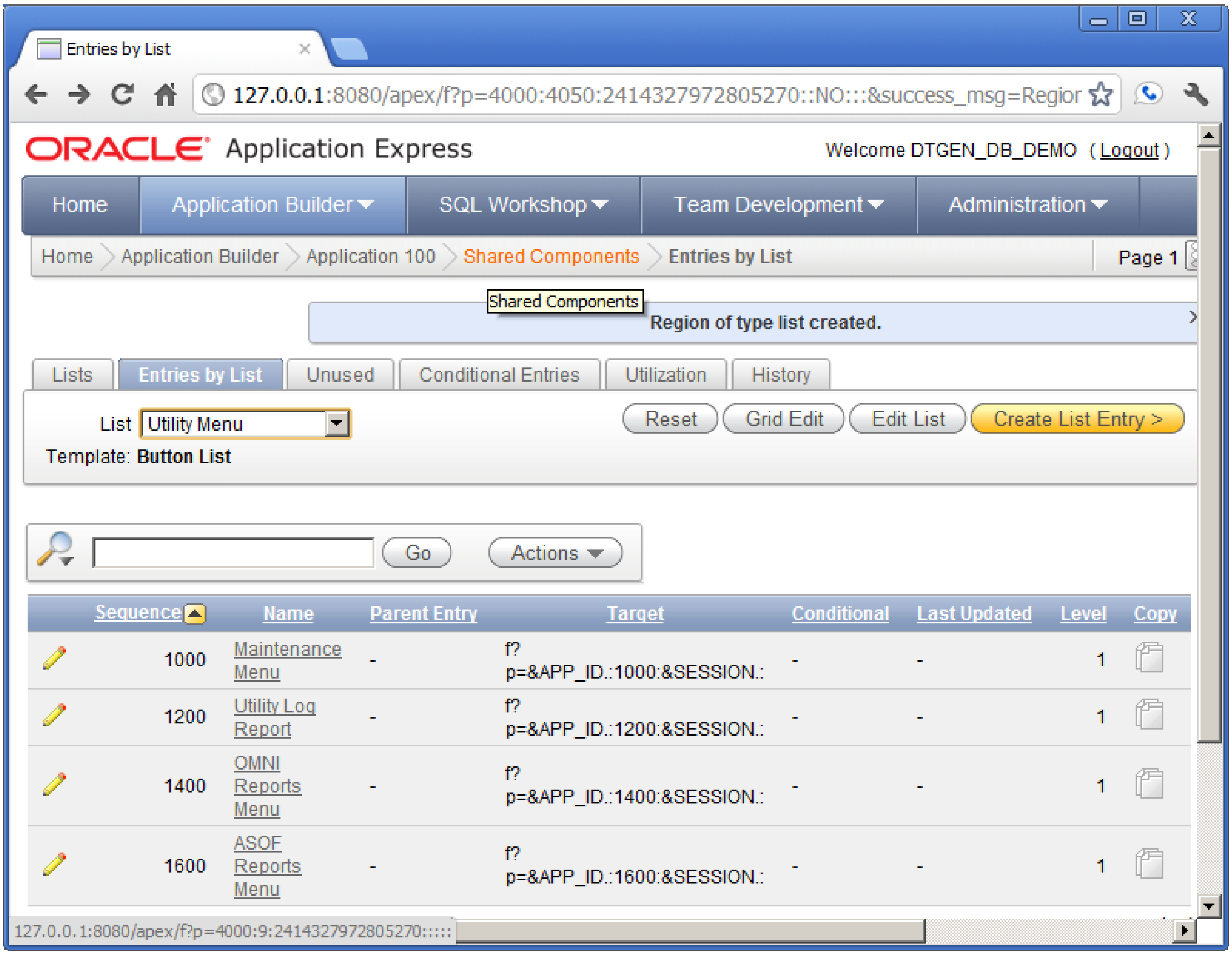
* Title	Utility Menu
Region Template	Chart List
Parent Region	- Select a Parent -
Display Point	Page Template Body (3. items above region content)
	[Body] [Pos.1] [Pos.2] [Pos.3] [Pos.4] [Pos.5]
* Sequence	20
Column	1

Confirm these values and click the "Next" button


Create Region
127.0.0.1:8080/apex/f?p=4000:407:2569188416850231::NO:::
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Application 100 > Page 1 > Create Region
Page 1
Create
Region
Display Attributes
Create Region
Cancel
< Previous
Next >
Create List Region
Recently Created or Edited Lists

* List: Utility Menu

Confirm these values and click the "Create List Region" button



Entries by List
127.0.0.1:8080/apex/f?p=4000:4050:2414327972805270::NO:::&success_msg=Regior
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Application 100 > Shared Components > Entries by List
Page 1
Shared Components

Region of type list created.

Lists
Entries by List
Unused
Conditional Entries
Utilization
History
Reset
Grid Edit
Edit List
Create List Entry >
Go
Actions

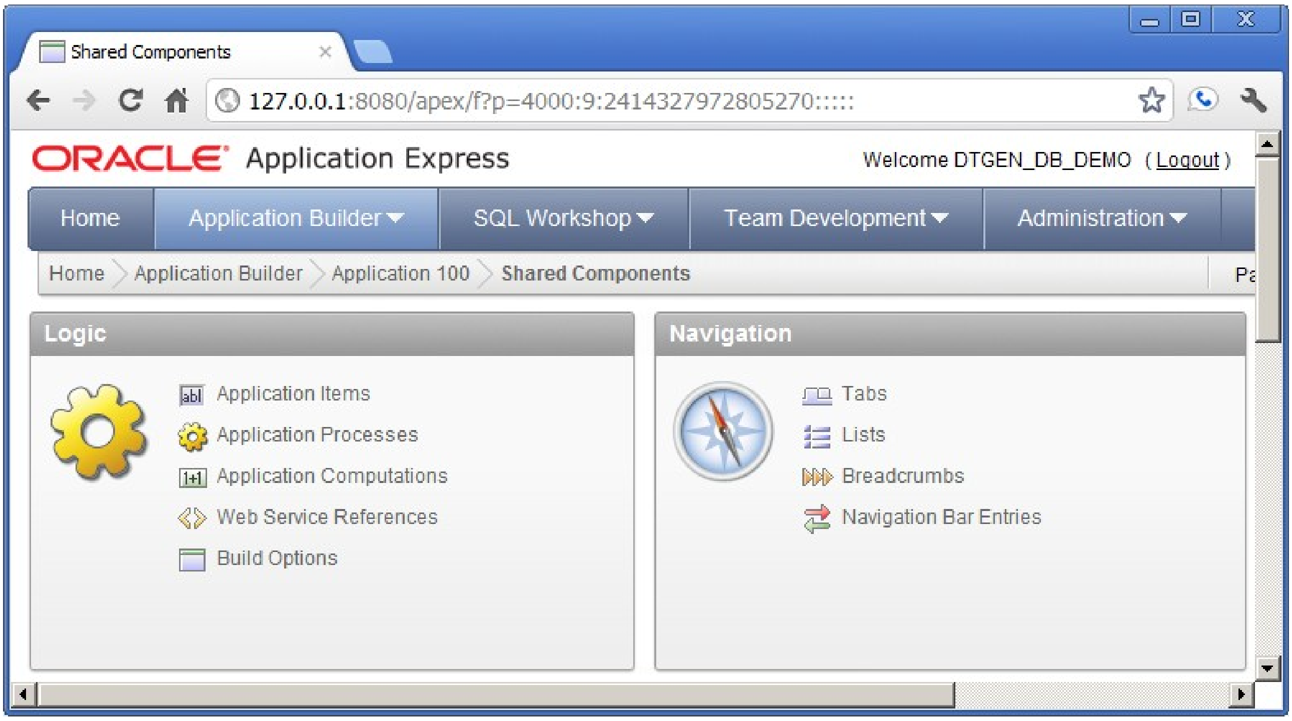
List: Utility Menu

Template: **Button List**

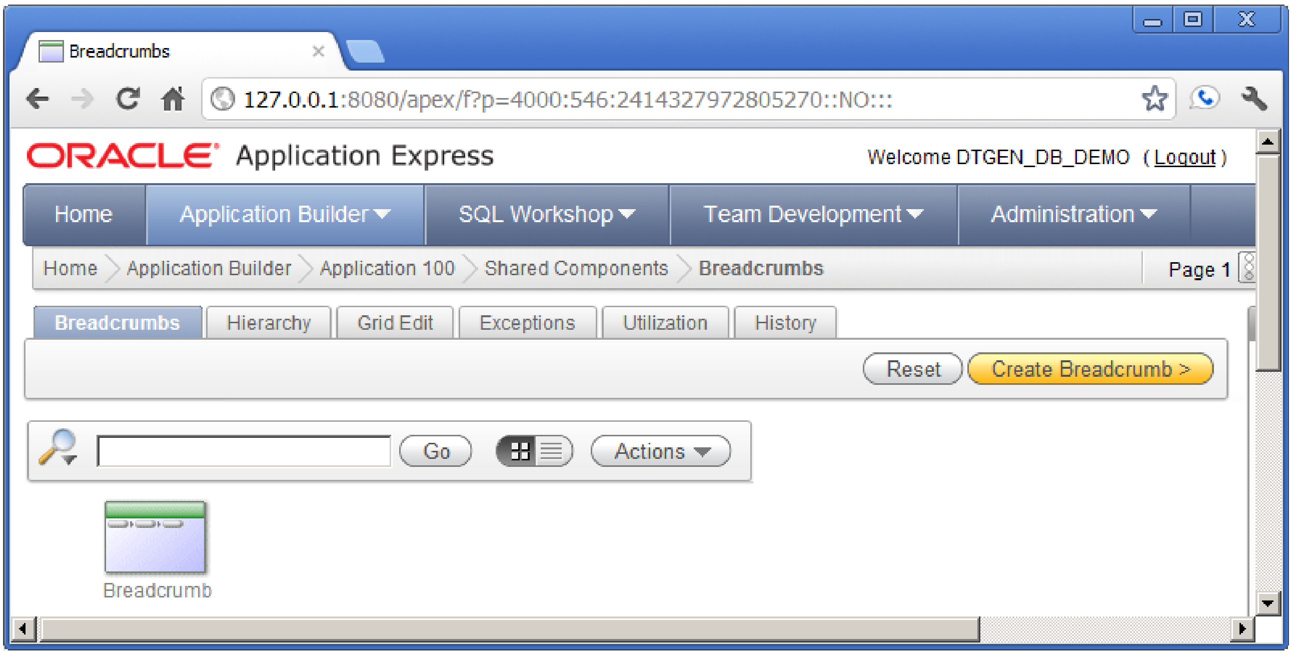
Sequence	Name	Parent Entry	Target	Conditional	Last Updated	Level	Copy
1000	Maintenance Menu	-	f?p=&APP_ID.:1000:&SESSION.:	-	-	1	
1200	Utility Log Report	-	f?p=&APP_ID.:1200:&SESSION.:	-	-	1	
1400	OMNI Reports Menu	-	f?p=&APP_ID.:1400:&SESSION.:	-	-	1	
1600	ASOF Reports Menu	-	f?p=&APP_ID.:1600:&SESSION.:	-	-	1	

127.0.0.1:8080/apex/f?p=4000:9:2414327972805270::::

Click the "Shared Components" link in the breadcrumb at the top.

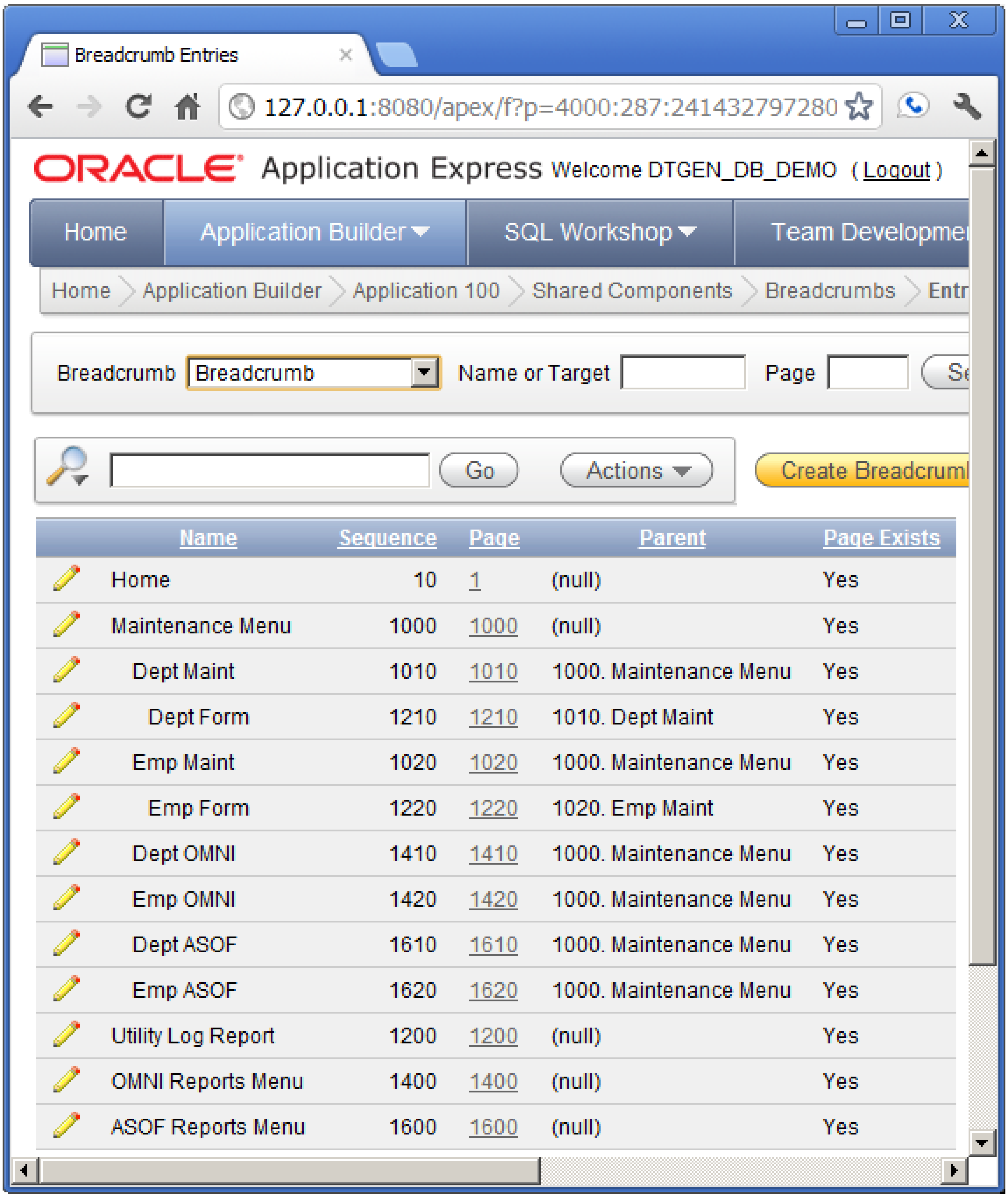


Click the "Breadcrumbs" link in the "Navigation" pane.



Breadcrumbs
127.0.0.1:8080/apex/f?p=4000:546:2414327972805270::NO:::
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Application 100 > Shared Components > Breadcrumbs
Page 1
Breadcrumbs
Hierarchy
Grid Edit
Exceptions
Utilization
History
Reset
Create Breadcrumb >
Go
Actions
Breadcrumb

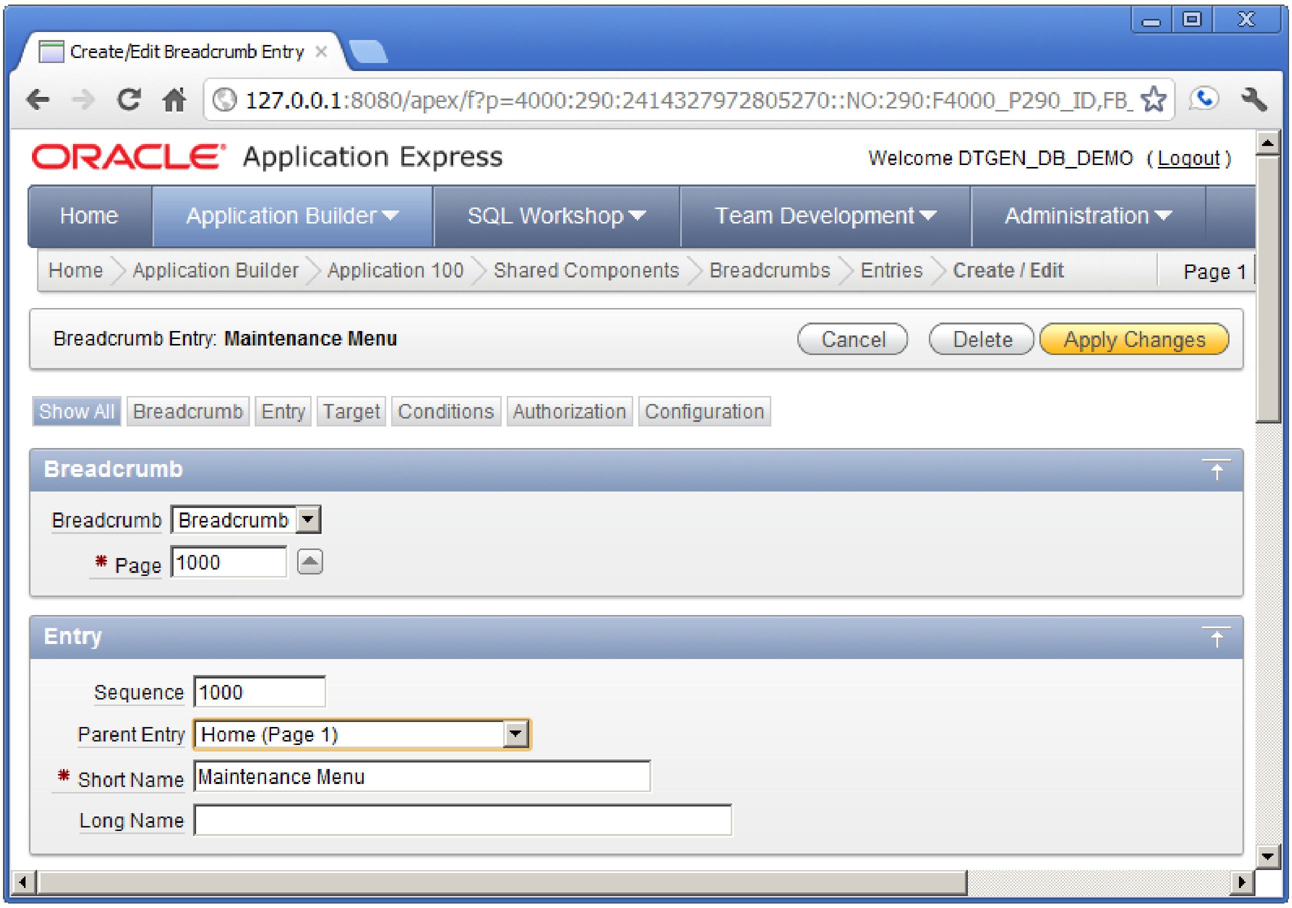
Click the "Breadcumb" Icon at the bottom left


Breadcrumb Entries
127.0.0.1:8080/apex/f?p=4000:287:241432797280
ORACLE Application Express Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Developme
Home > Application Builder > Application 100 > Shared Components > Breadcrumbs > Entr
Breadcrumb
Name or Target
Page
Se
Go
Actions
Create Breadcrum

Breadcrumb: Breadcrumb

Name	Sequence	Page	Parent	Page Exists
Home	10	1	(null)	Yes
Maintenance Menu	1000	1000	(null)	Yes
Dept Maint	1010	1010	1000. Maintenance Menu	Yes
Dept Form	1210	1210	1010. Dept Maint	Yes
Emp Maint	1020	1020	1000. Maintenance Menu	Yes
Emp Form	1220	1220	1020. Emp Maint	Yes
Dept OMNI	1410	1410	1000. Maintenance Menu	Yes
Emp OMNI	1420	1420	1000. Maintenance Menu	Yes
Dept ASOF	1610	1610	1000. Maintenance Menu	Yes
Emp ASOF	1620	1620	1000. Maintenance Menu	Yes
Utility Log Report	1200	1200	(null)	Yes
OMNI Reports Menu	1400	1400	(null)	Yes
ASOF Reports Menu	1600	1600	(null)	Yes

Click the pencil icon to the left of the "Maintenance Menu" name.



Create/Edit Breadcrumb Entry
127.0.0.1:8080/apex/f?p=4000:290:2414327972805270::NO:290:F4000_P290_ID,FB_
ORACLE Application Express
Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Administration
Home > Application Builder > Application 100 > Shared Components > Breadcrumbs > Entries > Create / Edit
Page 1
Cancel
Delete
Apply Changes
Show All
Breadcrumb
Entry
Target
Conditions
Authorization
Configuration

Breadcrumb Entry: **Maintenance Menu**

Breadcrumb

Breadcrumb	Breadcrumb
* Page	1000

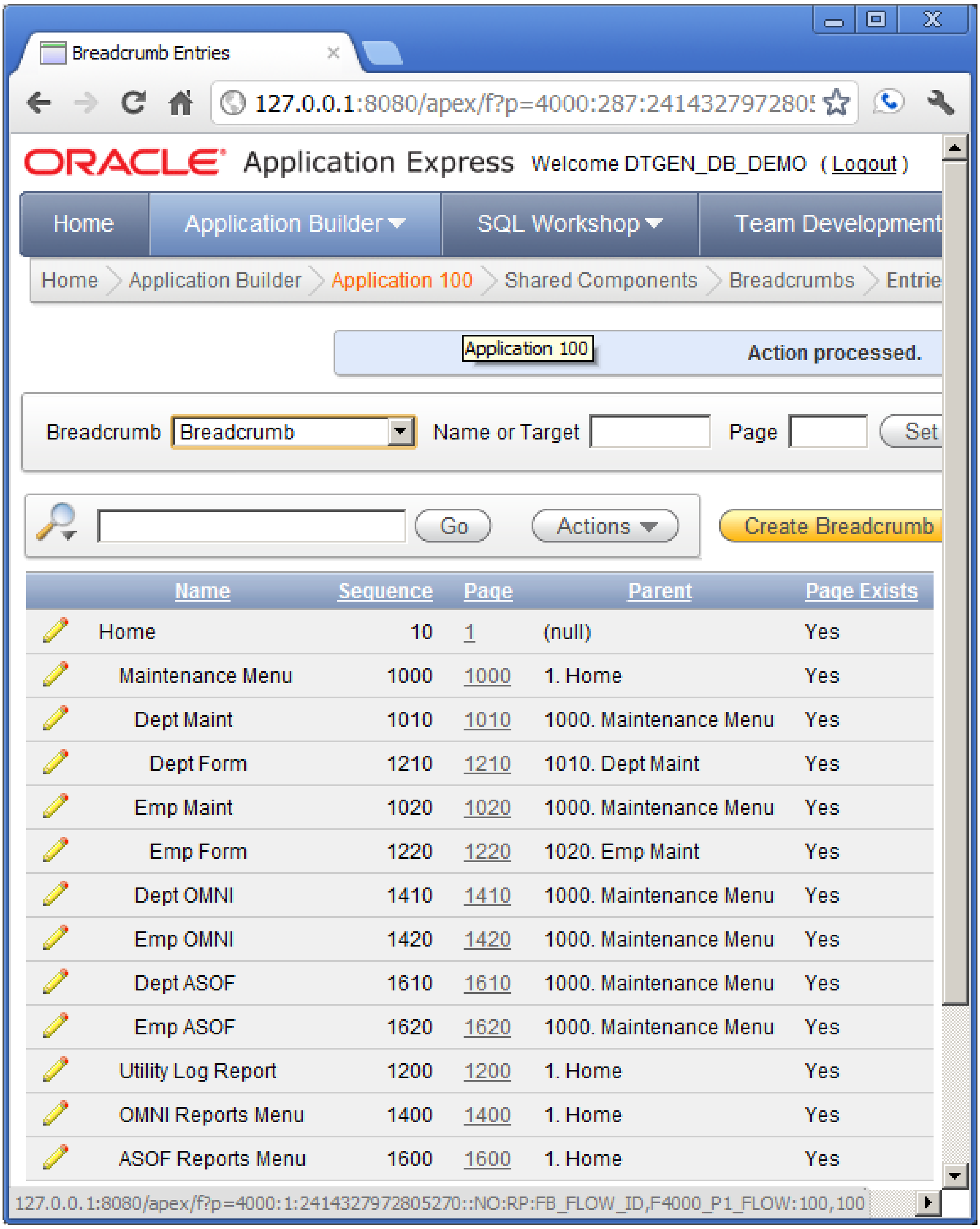
Entry

Sequence	1000
Parent Entry	Home (Page 1)
* Short Name	Maintenance Menu
Long Name	

Change the "Parent Entry" to "Home (Page 1)", then click "Apply Changes".

Repeat the above step for the following names:

- Utility Log Report
- OMNI Reports Menu
- ASOF Reports Menu

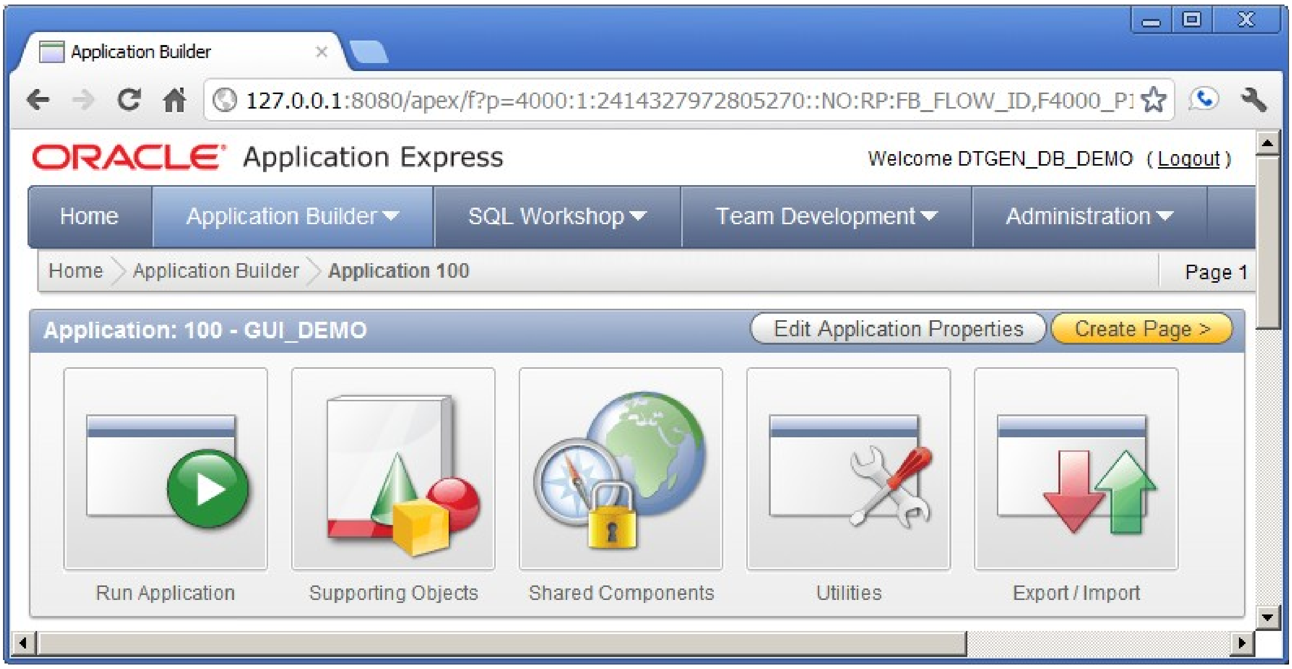


Breadcrumb Entries
127.0.0.1:8080/apex/f?p=4000:287:241432797280
ORACLE Application Express Welcome DTGEN_DB_DEMO (Logout)
Home
Application Builder
SQL Workshop
Team Development
Home > Application Builder > Application 100 > Shared Components > Breadcrumbs > Entrie
Application 100
Action processed.
Breadcrumb
Name or Target
Page
Set
Go
Actions
Create Breadcrumb
127.0.0.1:8080/apex/f?p=4000:1:2414327972805270::NO:RP:FB_FLOW_ID,F4000_P1_FLOW:100,100

Breadcrumb: Breadcrumb

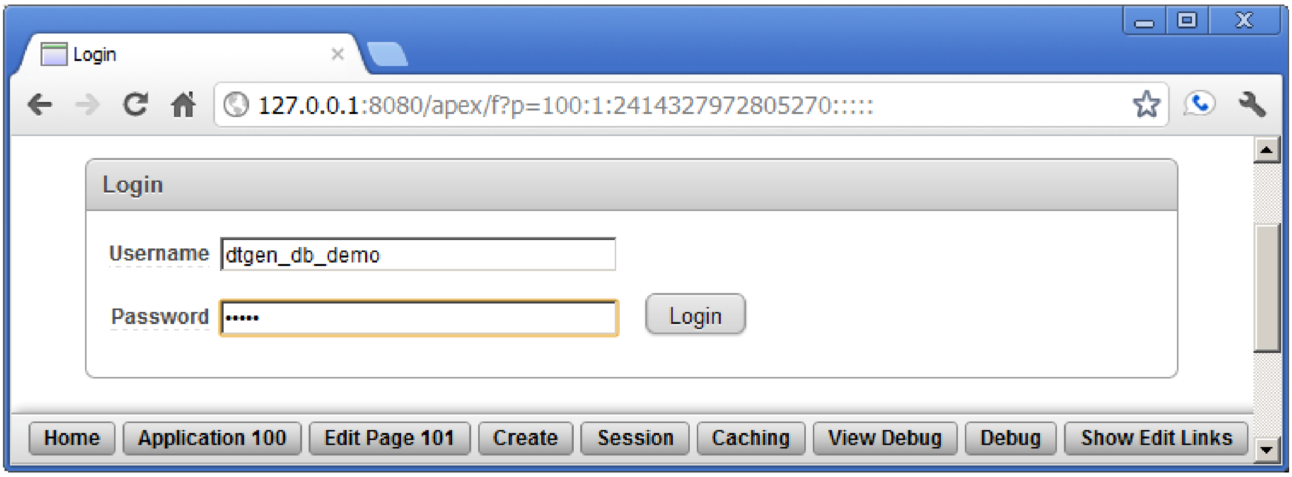
Name	Sequence	Page	Parent	Page Exists
Home	10	1	(null)	Yes
Maintenance Menu	1000	1000	1. Home	Yes
Dept Maint	1010	1010	1000. Maintenance Menu	Yes
Dept Form	1210	1210	1010. Dept Maint	Yes
Emp Maint	1020	1020	1000. Maintenance Menu	Yes
Emp Form	1220	1220	1020. Emp Maint	Yes
Dept OMNI	1410	1410	1000. Maintenance Menu	Yes
Emp OMNI	1420	1420	1000. Maintenance Menu	Yes
Dept ASOF	1610	1610	1000. Maintenance Menu	Yes
Emp ASOF	1620	1620	1000. Maintenance Menu	Yes
Utility Log Report	1200	1200	1. Home	Yes
OMNI Reports Menu	1400	1400	1. Home	Yes
ASOF Reports Menu	1600	1600	1. Home	Yes

After repeating the steps, confirm values and click the "Application 100" link in the breadcrumb at the top.



127.0.0.1:8080/apex/f?p=4000:1:2414327972805270::NO:RP:FB_FLOW_ID,F4000_P1

Click the "Run Application" button.

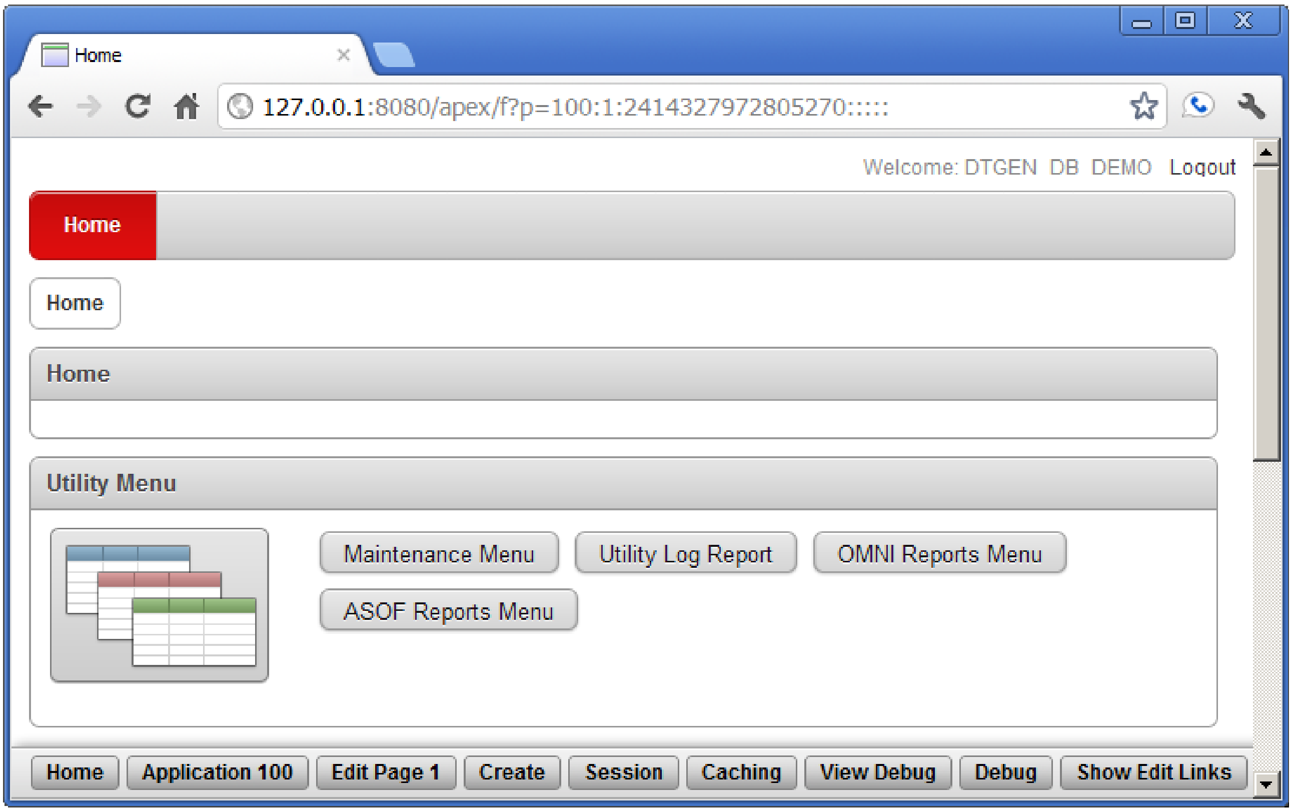


127.0.0.1:8080/apex/f?p=100:1:2414327972805270:::::

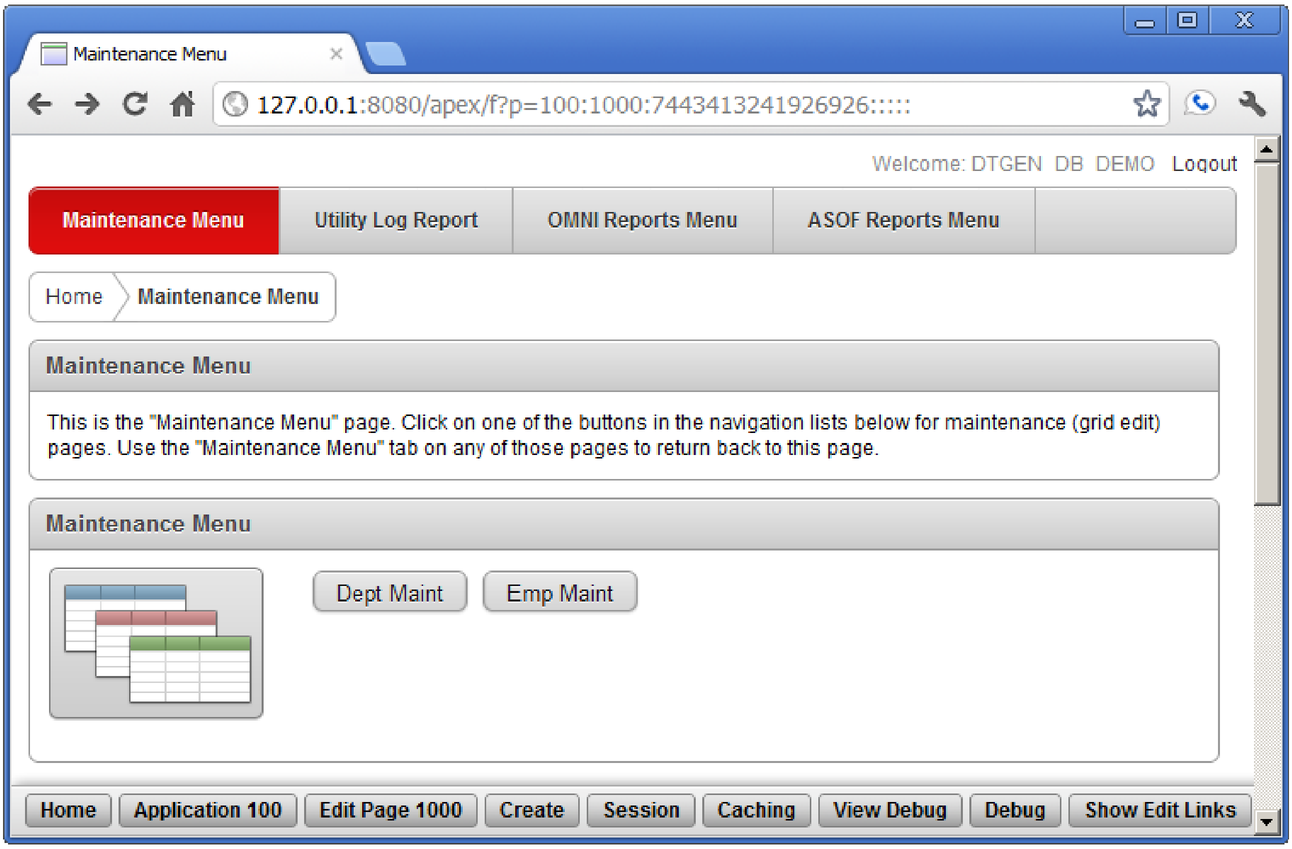
Username dtgen_db_demo

Password •••••

Enter the password "dtgen" and click the "Login" button.

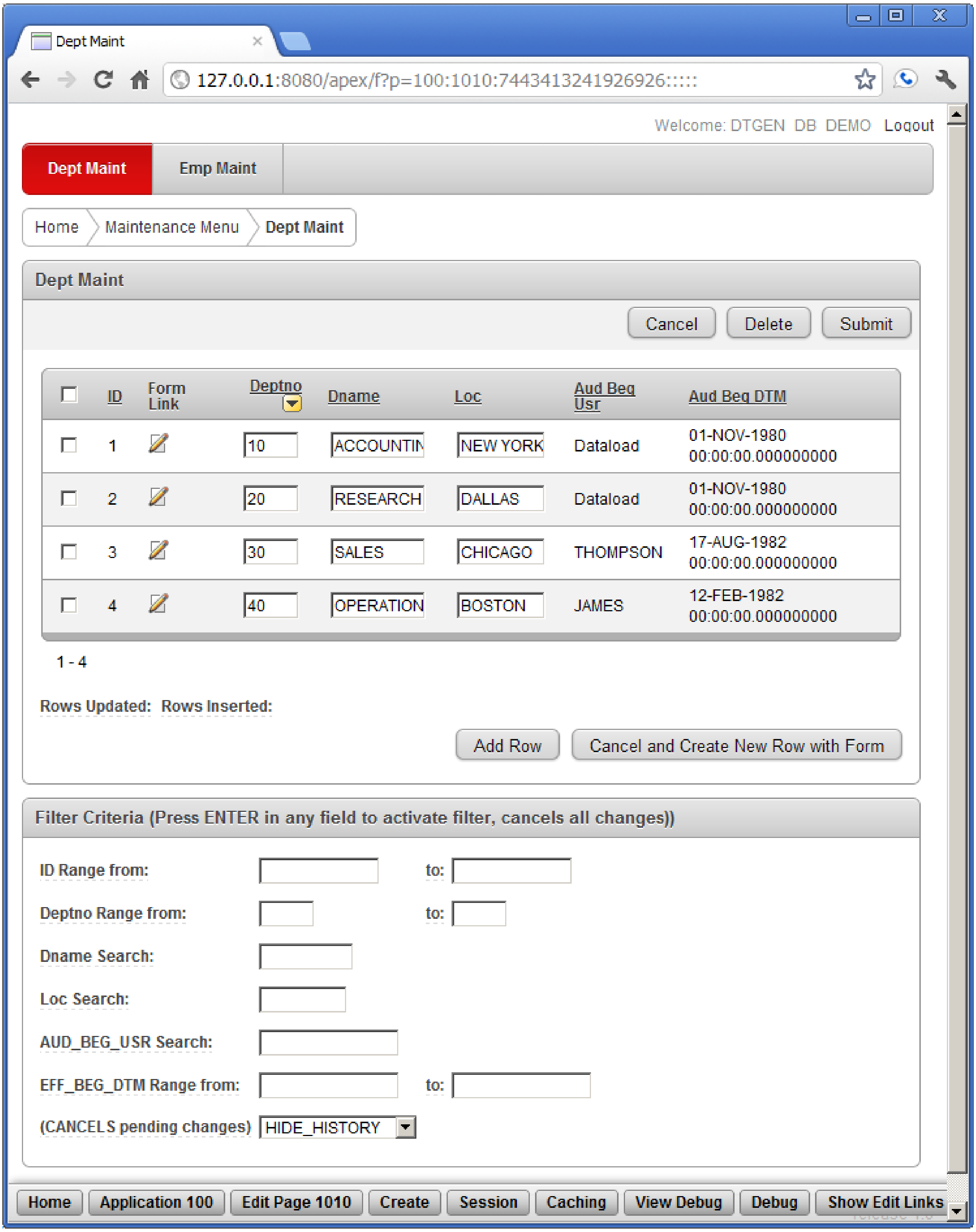


Click the "Maintenance Menu" button in the "Utility Menu" pane.



This is the "Maintenance Menu" page. Click on one of the buttons in the navigation lists below for maintenance (grid edit) pages. Use the "Maintenance Menu" tab on any of those pages to return back to this page.

Click the "Dept Maint" button in the "Maintenance Meny" pane.



Dept Maint

□	ID	Form Link	Deptno	Dname	Loc	Aud Beg Usr	Aud Beg DTM
□	1	✎	10	ACCOUNTIN	NEW YORK	Dataload	01-NOV-1980 00:00:00.000000000
□	2	✎	20	RESEARCH	DALLAS	Dataload	01-NOV-1980 00:00:00.000000000
□	3	✎	30	SALES	CHICAGO	THOMPSON	17-AUG-1982 00:00:00.000000000
□	4	✎	40	OPERATION	BOSTON	JAMES	12-FEB-1982 00:00:00.000000000

1 - 4

Rows Updated: Rows Inserted:

Filter Criteria (Press ENTER in any field to activate filter, cancels all changes))

ID Range from: to:

Deptno Range from: to:

Dname Search:

Loc Search:

AUD_BEG_USR Search:

EFF_BEG_DTM Range from: to:

(CANCELS pending changes) HIDE_HISTORY

This is the DEPT data maintenance form for the new application in APEX. This is not a completed application. These data maintenance forms and reports are not expected to be used by the application user. They are expected to become part of a super-user or application adminstrator's toolset, if they are used in the application. Customized forms that are more user-friendly should be added to this application before it is released to the application users.

Exercise #2: Data Domain Filterable Grid Edit

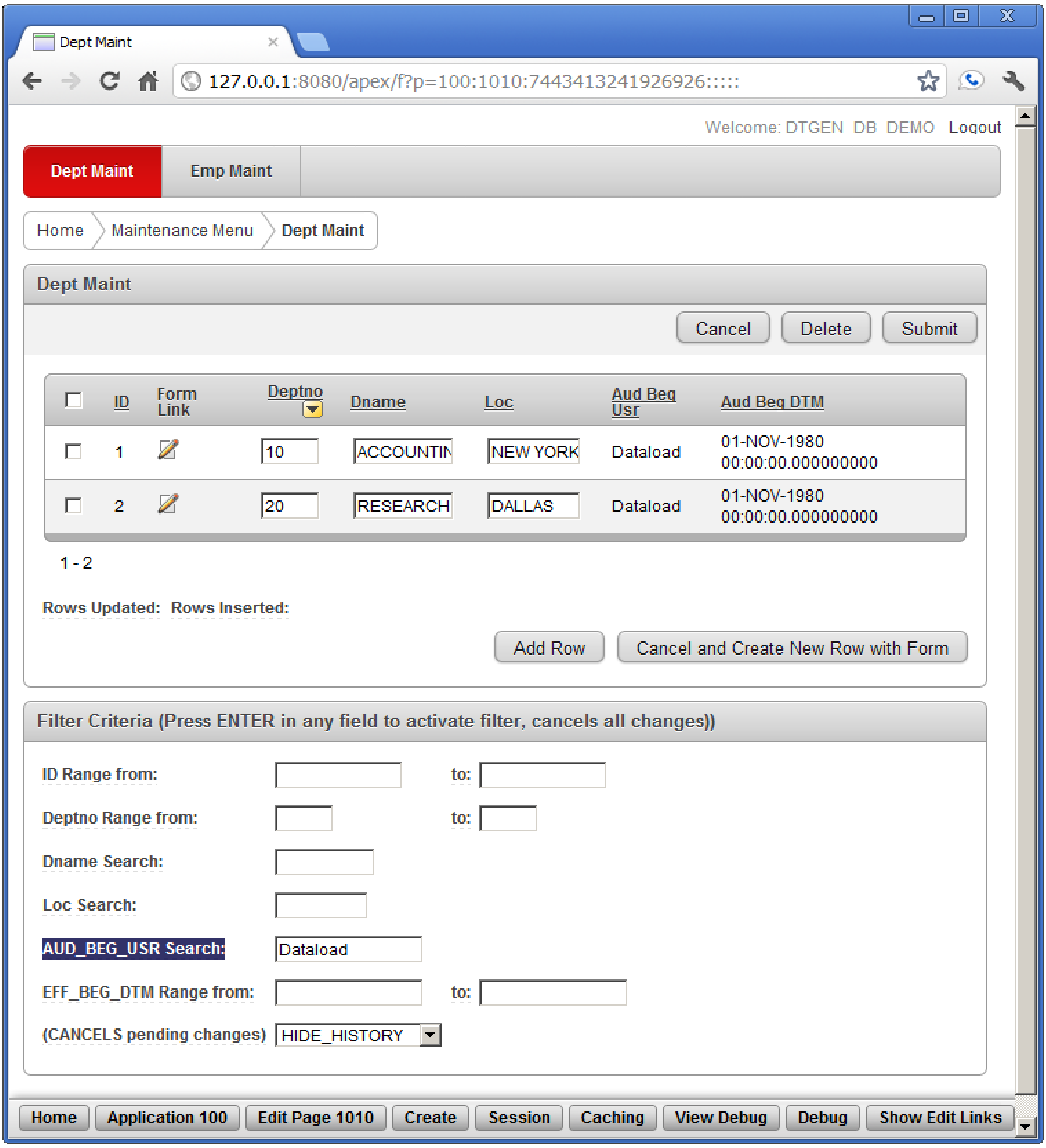
Command Line:

```
    sqlplus /nolog @e2
```

Exercise #2 does not modify the database. This exercise can be repeated without problem.

Exercise #1 finished with a DEPT maintenance form display in the APEX GUI. This exercise starts at that point. The data domain filterable grid edit is shown in the maintenance form. The grid edit portion is in the "Dept Maint" pane. Investigation of the grid edit is not pursued in detail in this exercise. Most of the fields and buttons in the "Dept Maint" pane should be self-evident. All the data sorting and modification should be functional. This is basically the grid edit functionality that APEX offers when creating a new page in an application.

The data domain filter is the "Filter Critera" pane at the bottom of the form. This filter is custom generated for each data type of each data field. As an example of how the filter works, enter "Dataload" in the "**AUD_BEG_USR Search:**" field and press ENTER. The screen should look like the following:


Dept Maint
127.0.0.1:8080/apex/f?p=100:1010:7443413241926926:::::
Welcome: DTGEN DB DEMO Logout
Dept Maint
Emp Maint
Home
Maintenance Menu
Dept Maint

Dept Maint

□	ID	Form Link	Deptno	Dname	Loc	Aud Beg Usr	Aud Beg DTM
□	1	✎	10	ACCOUNTIN	NEW YORK	Dataload	01-NOV-1980 00:00:00.000000000
□	2	✎	20	RESEARCH	DALLAS	Dataload	01-NOV-1980 00:00:00.000000000

1 - 2

Rows Updated: Rows Inserted:

Add Row
Cancel and Create New Row with Form

Filter Criteria (Press ENTER in any field to activate filter, cancels all changes))

ID Range from: to:

Deptno Range from: to:

Dname Search:

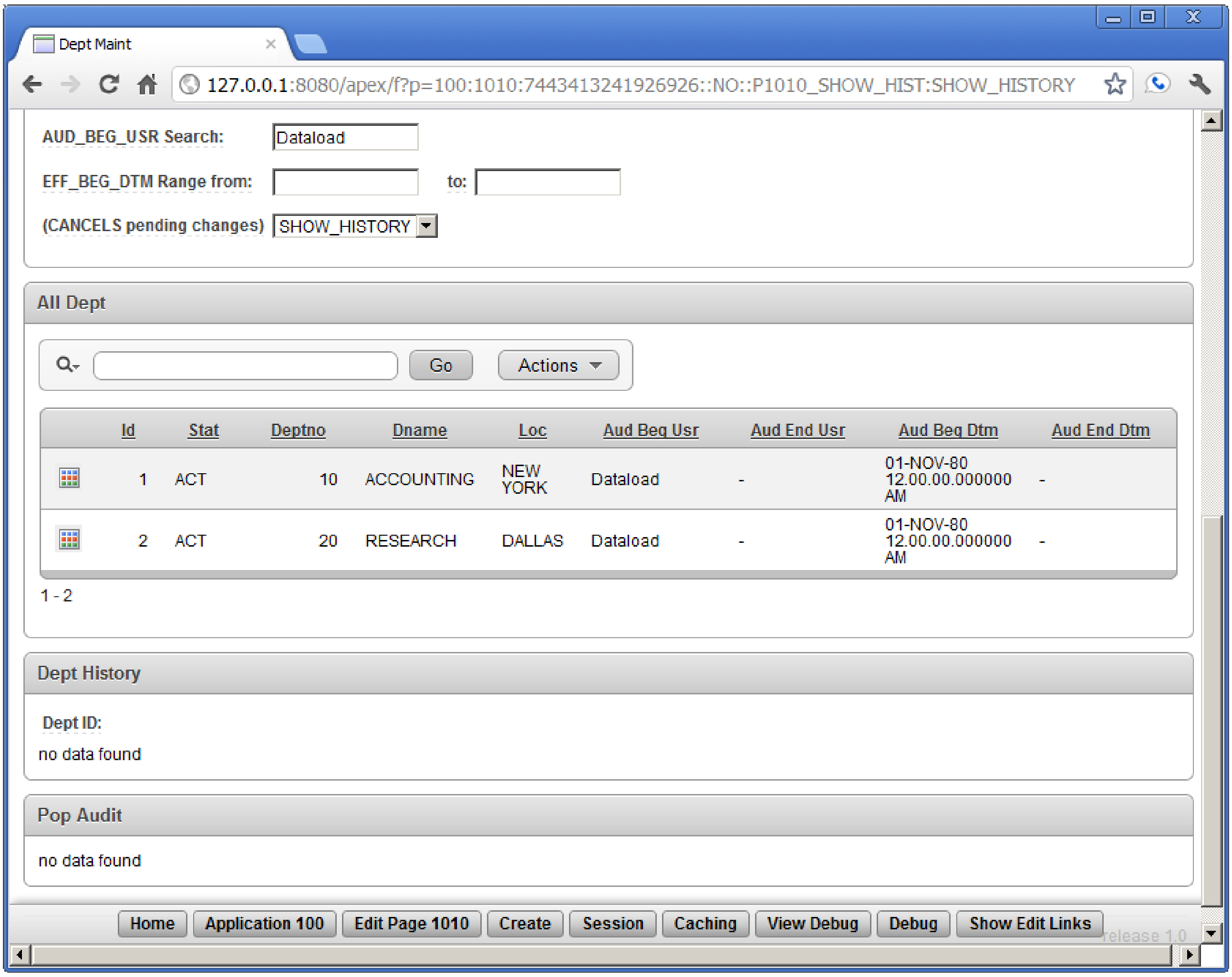
Loc Search:

AUD_BEG_USR Search: Dataload

EFF_BEG_DTM Range from: to:

(CANCELS pending changes) HIDE_HISTORY

The rows of the grid edit have been reduced to only the rows with an "Audit Begin User" of "Dataload". Select "Show History" at the bottom of the page to get this view at the bottom of the form:



AUD_BEG_USR Search: Dataload

EFF_BEG_DTM Range from: to:

(CANCELS pending changes) SHOW_HISTORY

All Dept

	Id	Stat	Deptno	Dname	Loc	Aud Beg Usr	Aud End Usr	Aud Beg Dtm	Aud End Dtm
▦	1	ACT	10	ACCOUNTING	NEW YORK	Dataload	-	01-NOV-80 12.00.00.000000 AM	-
▦	2	ACT	20	RESEARCH	DALLAS	Dataload	-	01-NOV-80 12.00.00.000000 AM	-

1 - 2

Dept History

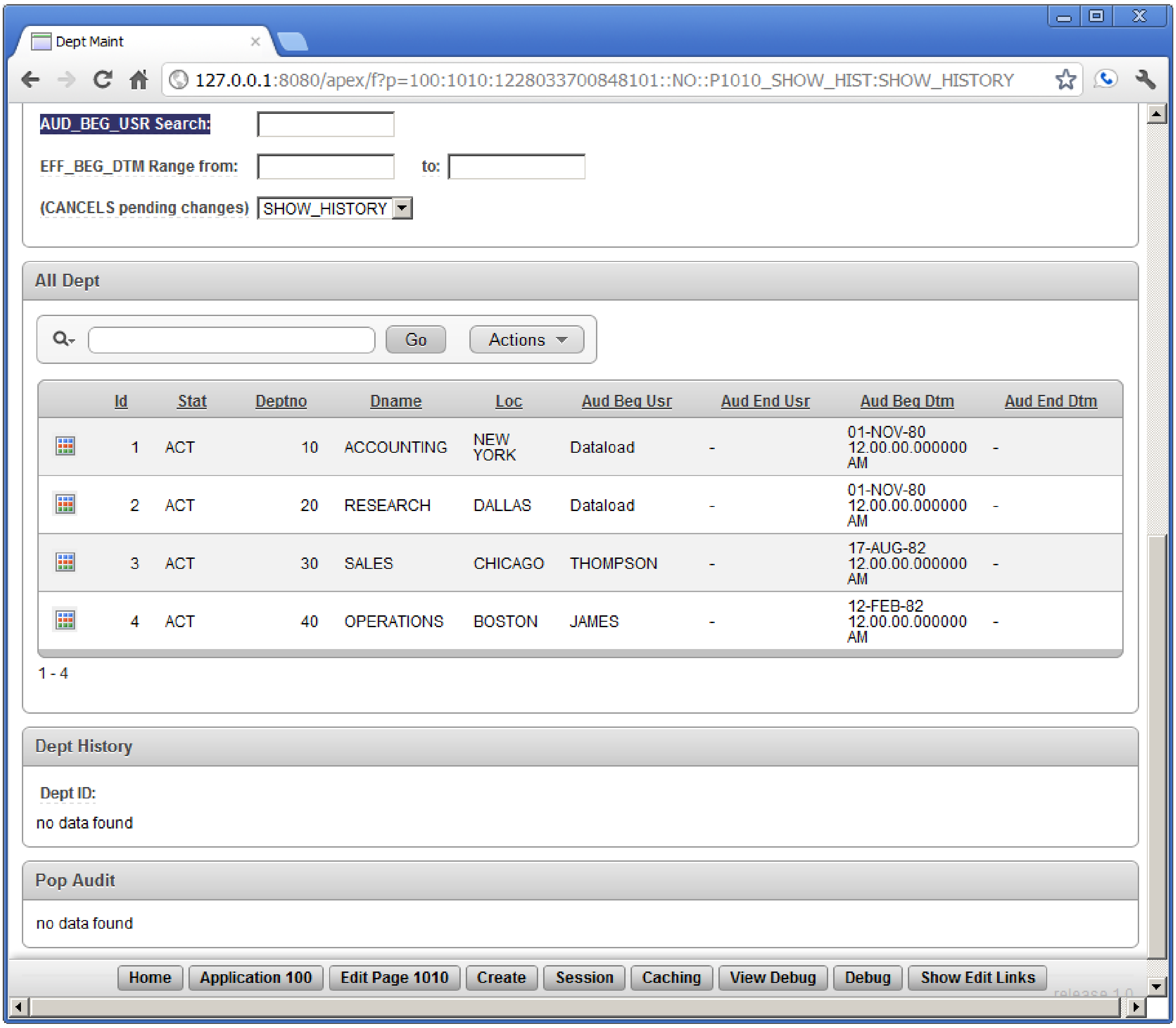
Dept ID:

no data found

Pop Audit

no data found

In this bottom part of the form, the "All Instances" view of current and historical data is contained in the "All Dept" pane. This is a standard APEX Interactive Report. Additional filters and sorting can be done via the search bar. Note that the interactive report is filtered by the current limitation on "Audit Begin User". Clear that field and press ENTER to get the following screen:


Dept Maint
127.0.0.1:8080/apex/f?p=100:1010:1228033700848101::NO::P1010_SHOW_HIST:SHOW_HISTORY

AUD_BEG_USR Search:

EFF_BEG_DTM Range from: to:

(CANCELS pending changes) SHOW_HISTORY

All Dept

	Id	Stat	Deptno	Dname	Loc	Aud Beg Usr	Aud End Usr	Aud Beg Dtm	Aud End Dtm
▦	1	ACT	10	ACCOUNTING	NEW YORK	Dataload	-	01-NOV-80 12.00.00.000000 AM	-
▦	2	ACT	20	RESEARCH	DALLAS	Dataload	-	01-NOV-80 12.00.00.000000 AM	-
▦	3	ACT	30	SALES	CHICAGO	THOMPSON	-	17-AUG-82 12.00.00.000000 AM	-
▦	4	ACT	40	OPERATIONS	BOSTON	JAMES	-	12-FEB-82 12.00.00.000000 AM	-

1 - 4

Dept History

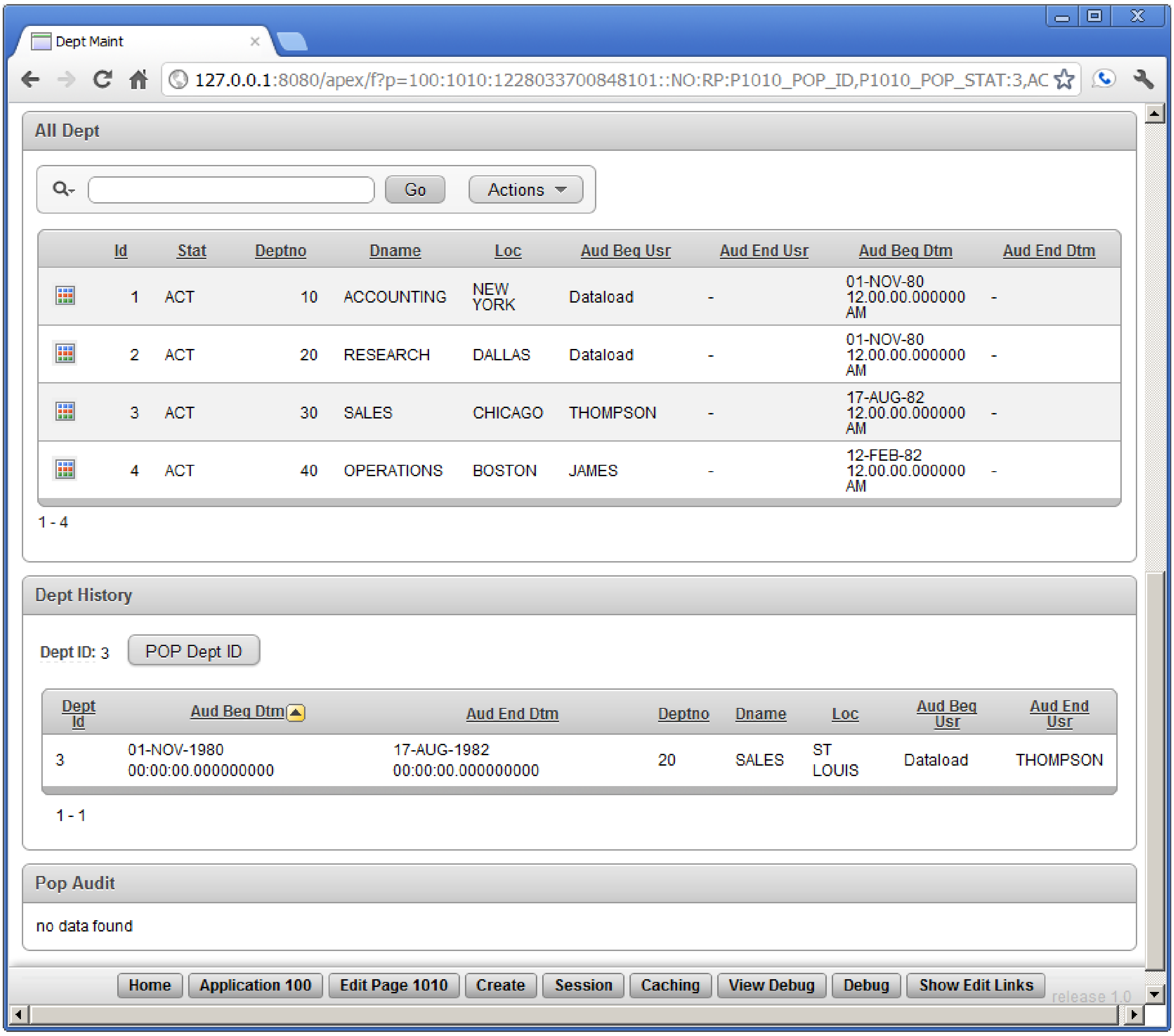
Dept ID:

no data found

Pop Audit

no data found

Notice that all the active records appear in the "All Dept" pane, similar to the records in the "Dept Maint" pane (not shown). There are no departments that have been "deleted", so there are no current instances of a department in history (i.e. STAT column has all 'ACT' and no 'AUD'). Click the gird icon at the left side of the SALES record in the "ALL Dept" pane.



All Dept

	Id	Stat	Deptno	Dname	Loc	Aud Beg Usr	Aud End Usr	Aud Beg Dtm	Aud End Dtm
▦	1	ACT	10	ACCOUNTING	NEW YORK	Dataload	-	01-NOV-80 12.00.00.000000 AM	-
▦	2	ACT	20	RESEARCH	DALLAS	Dataload	-	01-NOV-80 12.00.00.000000 AM	-
▦	3	ACT	30	SALES	CHICAGO	THOMPSON	-	17-AUG-82 12.00.00.000000 AM	-
▦	4	ACT	40	OPERATIONS	BOSTON	JAMES	-	12-FEB-82 12.00.00.000000 AM	-

1 - 4

Dept History

Dept ID: 3

Dept Id	Aud Beg Dtm	Aud End Dtm	Deptno	Dname	Loc	Aud Beg Usr	Aud End Usr
3	01-NOV-1980 00:00:00.000000000	17-AUG-1982 00:00:00.000000000	20	SALES	ST LOUIS	Dataload	THOMPSON

1 - 1

Pop Audit

no data found

Selecting the SALES department instance in the "All Dept" pane shows the history of the SALES department in the "Dept History" pane. The sales department was in St. Loius from November 1980 before moving to Chicago in August 1982. Click the "POP Dept ID" button in the "All Dept" pane.

Pressing this button would "UNDO" the sales department move to Chicago and create

Exercise #3: Handling of CLOB data

Command Line:

```
    sqlplus /nolog @e3
```

Exercise #3 does not modify the database. This exercise can be repeated without problem.

In this exercise, we take a trip back in time using the DTGen version of flashback query. The ASOF view on each table returns data that was current during the "util.set_asof_dtm" date/time specified. Referential integrity is not specifically enforced in the AUD and HIST data. However, since referential integrity was enforced at any given time, the data for any point in time should have integrity. (NOTE: Referential integrity is not guaranteed until Issue #2 is resolved. See "http://code.google.com/p/dtgen/issues/detail?id=2 ")

The following query joins the EMP_ASOF and DEPT_ASOF views at a point in time of January

1st, 1983 (midnight).

```
SQL>
SQL> execute glob.set_asof_dtm(to_timestamp('1983-01-01', 'YYYY-MM-DD'))

PL/SQL procedure successfully completed.

SQL>
SQL> select empno, ename, job, mgr_emp_nk1, hiredate, sal, deptno, dname, loc
  2   from  emp_asof e, dept_asof d where e.dept_id = d.id
  3   order by empno;

EMPNO ENAME    JOB       MGR_EMP_NK1 HIREDATE    SAL DEPTNO DNAME      LOC
----- -------- --------- ----------- --------- ----- ------ ---------- --------
 7369 SMITH    CLERK            7839 17-DEC-80   800     10 ACCOUNTING NEW YORK
 7499 ALLEN    SALESMAN         7698 20-FEB-81  1600     30 SALES      CHICAGO
 7521 WARD     SALESMAN         7698 22-FEB-81  1250     30 SALES      CHICAGO
 7566 JONES    MANAGER          7839 02-APR-81  2975     20 RESEARCH   DALLAS
 7654 MARTIN   SALESMAN         7698 28-SEP-81  1250     30 SALES      CHICAGO
 7698 BLAKE    MANAGER          7839 01-MAY-81  2850     30 SALES      CHICAGO
 7782 CLARK    MANAGER          7839 09-JUN-81  2450     10 ACCOUNTING NEW YORK
 7788 SCOTT    ANALYST          7566 09-DEC-82  3000     20 RESEARCH   DALLAS
 7839 KING     PRESIDENT             17-NOV-81  5000     10 ACCOUNTING NEW YORK
 7844 TURNER   SALESMAN         7698 08-SEP-81  1500     30 SALES      CHICAGO
 7900 JAMES    CLERK            7698 03-DEC-81   950     30 SALES      CHICAGO
 7902 FORD     ANALYST          7566 03-DEC-81  3000     20 RESEARCH   DALLAS
 7934 MILLER   CLERK            7782 23-JAN-82  1300     10 ACCOUNTING NEW YORK

13 rows selected.
```

Data from the original dmobld.sql script showed that ADAMS the CLERK was hired on January 12th, 1983. Since the query above is for data some 11 days earlier, ADAMS is missing from the list. The following is the same query with "asof_dtm" set one year earler.

```
SQL>
SQL> execute glob.set_asof_dtm(to_timestamp('1982-01-01', 'YYYY-MM-DD'))

PL/SQL procedure successfully completed.

SQL>
SQL> select empno, ename, job, mgr_emp_nk1, hiredate, sal, deptno, dname, loc
  2   from  emp_asof e, dept_asof d where e.dept_id = d.id
  3   order by empno;

EMPNO ENAME    JOB       MGR_EMP_NK1 HIREDATE    SAL DEPTNO DNAME      LOC
----- -------- --------- ----------- --------- ----- ------ ---------- --------
 7369 SMITH    CLERK            7839 17-DEC-80   800     10 ACCOUNTING NEW YORK
 7499 ALLEN    SALESMAN         7698 20-FEB-81  1600     20 SALES      ST LOUIS
 7521 WARD     SALESMAN         7698 22-FEB-81  1250     20 SALES      ST LOUIS
 7566 JONES    MANAGER          7839 02-APR-81  2975     20 RESEARCH   DALLAS
 7654 MARTIN   SALESMAN         7698 28-SEP-81  1250     20 SALES      ST LOUIS
 7698 BLAKE    MANAGER          7839 01-MAY-81  2850     20 SALES      ST LOUIS
 7782 CLARK    MANAGER          7839 09-JUN-81  2450     10 ACCOUNTING NEW YORK
 7788 SCOTT    ANALYST          7566 12-JUN-81  3000     20 RESEARCH   DALLAS
 7839 KING     PRESIDENT             17-NOV-81  5000     10 ACCOUNTING NEW YORK
 7844 TURNER   SALESMAN         7698 08-SEP-81  1500     20 SALES      ST LOUIS
 7876 ADAMS    CLERK            7566 22-NOV-81  1100     20 RESEARCH   DALLAS
 7900 JAMES    CLERK            7698 03-DEC-81   950     20 SALES      ST LOUIS
 7902 FORD     ANALYST          7566 03-DEC-81  3000     20 RESEARCH   DALLAS

13 rows selected.
```

Notice is that the SALES department is in St. Louis. Sometime during 1982, the SALES department must have been moved to Chicago. With a HIREDATE of January 23rd, 1982, MILLER is not on the list. However, ADAMS the CLERK is back on the list. Given the sequential assignment of EMPNO 7876, ADAMS was hired, left, and re-hired with the same EMPNO (but a new EMP record). More interesting data can be seen on September 1st, 1981 and June 1st, 1981.

Exercise #4: Comprehensive OMNI View Forms

Command Line:

```
sqlplus /nolog @e4
```

Exercise #4 modifies the database. The "drop_demo_users.sql", "create_demo_users.sql", and "e1.sql" scripts must be used to reset the database before re-running this exercise.

Each table defined in DTGen as a "LOG" or "EFF" table type has a "POP" or undo package generated for it. The POP function can undo all DML on a record. Below, department 40 will be staffed with new hires and transfers.

```
SQL>
SQL> select id did, deptno, dname, loc from dept_act;

DID     DEPTNO DNAME          LOC
--- ---------- -------------- -------------
  1         10 ACCOUNTING     NEW YORK
  2         20 RESEARCH       DALLAS
  3         30 SALES          CHICAGO
  4         40 OPERATIONS     BOSTON

4 rows selected.

SQL>
SQL> select empno, ename, job, dept_id did, dept_nk1 dept, aud_beg_usr, aud_beg_dtm
  2   from emp_act where dept_nk1 = 40;

no rows selected

SQL>
SQL> execute util.set_usr('SMITH');

PL/SQL procedure successfully completed.

SQL>
SQL> -- Add a new manager MCMURRY to the Operations Department
SQL> insert into emp_act (empno, ename, job, mgr_emp_nk1, hiredate, sal, dept_id)
  2   values (8156, 'MCMURRY', 'MANAGER', 7839, sysdate, 2975, 4);

1 row created.

SQL>
SQL> -- Add a new analyst WALKER to the Operations Department
SQL> insert into emp_act (empno, ename, job, mgr_emp_nk1, hiredate, sal, dept_nk1)
  2   values (8157, 'WALKER', 'ANALYST', 8156, sysdate, 3000, 40);

1 row created.

SQL>
SQL> -- Transfer an analyst SCOTT to the Operations Department
SQL> update emp_act
  2    set  dept_id    = 4
  3        ,mgr_emp_nk1 = 8156
  4   where empno = 7788;

1 row updated.

SQL>
SQL> -- Transfer a clerk JAMES to the Operations Department
SQL> update emp_act
  2    set  dept_nk1   = 40
  3        ,mgr_emp_nk1 = 8156
  4   where empno = 7902;

1 row updated.

SQL>
SQL> commit;
```

```
Commit complete.
```

In the sequence of steps above, department 40 is confirmed to have no employees. SMITH adds two new employees to the department and transfers 2 existing employees . The transaction is commited. However, something went wrong.

```
SQL>
SQL> select id eid, empno, ename, job, dept_id did, dept_nk1 dept,
  2        aud_beg_usr, aud_beg_dtm
  3   from emp_act where dept_nk1 = 40;

EID EMPNO ENAME    JOB       DID DEPT AUD_BEG AUD_BEG_D
--- ----- -------- --------- --- ---- ------- ---------
 18  7902 FORD     ANALYST     4   40 SMITH   18-APR-12
 20  7788 SCOTT    ANALYST     4   40 SMITH   18-APR-12
 23  8156 MCMURRY  MANAGER     4   40 SMITH   18-APR-12
 24  8157 WALKER   ANALYST     4   40 SMITH   18-APR-12

4 rows selected.
```

In the department listng above, FORD has been mistakenly transfered to department 40. MILLER will use the POP procedure to undo the "committed" error and re-transfer the correct employee.

```
SQL>
SQL> execute util.set_usr('MILLER');

PL/SQL procedure successfully completed.

SQL>
SQL> select emp_id eid, empno, ename, dept_id did,
  2        aud_beg_usr, aud_beg_dtm, aud_end_usr, aud_end_dtm
  3   from emp_hist where empno = 7902;

EID EMPNO ENAME    DID AUD_BEG AUD_BEG_D AUD_END AUD_END_D
--- ----- -------- --- ------- --------- ------- ---------
 18  7902 FORD       2 SMITH   02-DEC-81 SMITH   18-APR-12

1 row selected.

SQL>
SQL> -- Undo the transfer of FORD to the Operations Department
SQL> declare
  2    emp_id  number;
  3  begin
  4    select id into emp_id from emp_act where empno = 7902;
  5    emp_pop.at_server(emp_id);
  6  end;
  7  /

PL/SQL procedure successfully completed.

SQL>
SQL> -- Transfer a clerk JAMES to the Operations Department
SQL> update emp_act
  2    set  dept_nk1  = 40
  3       ,mgr_emp_nk1 = 8156
  4   where empno = 7900;

1 row updated.

SQL>
SQL> select id eid, empno, ename, job, dept_id did, dept_nk1 dept,
  2        aud_beg_usr, aud_beg_dtm
  3   from emp_act where dept_nk1 = 40 or empno = 7902;

EID EMPNO ENAME    JOB       DID DEPT AUD_BEG AUD_BEG_D
--- ----- -------- --------- --- ---- ------- ---------
 17  7900 JAMES    CLERK       4   40 MILLER  18-APR-12
 18  7902 FORD     ANALYST     2   20 SMITH   01-DEC-81
 20  7788 SCOTT    ANALYST     4   40 SMITH   18-APR-12
 23  8156 MCMURRY  MANAGER     4   40 SMITH   18-APR-12
 24  8157 WALKER   ANALYST     4   40 SMITH   18-APR-12

5 rows selected.

SQL>
SQL> select emp_id eid, empno, ename, dept_id did,
  2        aud_beg_usr, aud_beg_dtm, aud_end_usr, aud_end_dtm
  3   from emp_hist where empno = 7902;

no rows selected

SQL>
SQL> commit;

Commit complete.
```

In the first query above, FORD's history record is displayed, showing that he was in DEPT_ID 2. MILLER uses the EMP_POP.AT_SERVER procedures to "undo" the last DML on FORD's employment record. Then, MILLER correctly transfers JAMES to department 40 (DEPT_ID 4). Notice that the EMP_ACT record for FORD in the last query shows the original AUD_BEG_DTM of December 1st, 1981. Even though the EMP_ACT and EMP_HIST tables dont' show any record of the POP occuring, the POP was recorded in the EMP_PDAT table.

```
SQL>
SQL> select emp_id eid, pop_dml, pop_usr, pop_dtm, empno, ename,
  2        aud_beg_usr, aud_beg_dtm, aud_prev_beg_usr, aud_prev_beg_dtm
  3   from emp_pdat;

EID POP_DM POP_USR POP_DTM   EMPNO ENAME    AUD_BEG AUD_BEG_D AUD_PRE AUD_PREV_
--- ------ ------- --------- ----- -------- ------- --------- ------- ---------
 18 UPDATE MILLER  18-APR-12  7902 FORD     SMITH   18-APR-12 SMITH   01-DEC-81

1 row selected.
```

The query above shows that MILLER "popped" an UPDATE statement on FORD's employment record (ID 18) that was performed by SMITH on April 12th, 2012. Note that the AUD_PREV_BEG_USR of SMITH is the value restored to the AUD_BEG_USR in the EMP_ACT record. Also note that the AUD_PREV_BEG_DTM of December 1st, 1981 is the value restored to the AUD_BEG_DTM in the EMP_ACT record. These values, along with the ID from the original EMP_ACT record, can be used to trace the audit trail back to the original EMP_ACT or EMP_HIST records, regardless of how many times a record is "popped".

Exercise #5: Forms Development Guidelines

Command Line:

```
sqlplus /nolog @e5
```

Exercise #5 modifies the database. The "drop_demo_users.sql", "create_demo_users.sql", and "e1.sql" scripts must be used to reset the database before re-running this exercise.

Each record that was originally created in the EMP table is tracked through all the DML performed on it (including the use of the POP procedure). The original record is tracked via the ID it was originally assigned in the EMP table. In the EMP table, each record represents an instance of employment.

```
SQL>
SQL> select empno, ename, id eid, stat, dept_id did,
  2        aud_beg_usr, aud_beg_dtm, aud_end_usr, aud_end_dtm
  3   from emp_all order by empno, id;

EMPNO ENAME    EID STAT DID AUD_BEG_ AUD_BEG_D AUD_END_ AUD_END_D
----- -------- --- ---- --- -------- --------- -------- ---------
 7301 ELLISON    1 HIST   1 DAVIS    30-OCT-80 THOMPSON 30-JUN-81
 7344 DAVIS      2 HIST   1 SMITH    28-NOV-81 SMITH    10-DEC-81
 7369 SMITH      3 ACT    2 SMITH    28-FEB-83
 7499 ALLEN      4 ACT    3 THOMPSON 14-MAY-81
 7521 WARD       5 ACT    3 THOMPSON 14-MAY-81
 7566 JONES      6 ACT    2 SMITH    30-NOV-81
 7654 MARTIN     7 HIST   3 THOMPSON 18-APR-81 THOMPSON 14-MAY-81
 7654 MARTIN    14 ACT    3 SMITH    26-SEP-81
 7698 BLAKE      8 ACT    3 SMITH    28-NOV-81
 7782 CLARK      9 ACT    1 SMITH    01-DEC-81
 7788 SCOTT     10 HIST   2 THOMPSON 10-JUN-81 JAMES    07-MAR-82
 7788 SCOTT     20 ACT    4 SMITH    18-APR-12
 7839 KING      11 HIST   1 THOMPSON 14-JUN-81 SMITH    29-AUG-81
 7839 KING      15 ACT    1 SMITH    15-NOV-81
 7840 LANE      12 HIST   1 THOMPSON 15-AUG-81 SMITH    28-NOV-81
 7844 TURNER    13 ACT    3 SMITH    09-SEP-81
 7876 ADAMS     16 HIST   2 SMITH    24-NOV-81 JAMES    15-JUN-82
 7876 ADAMS     21 ACT    2 SMITH    09-JAN-83
 7900 JAMES     17 ACT    4 MILLER   18-APR-12
 7902 FORD      18 ACT    2 SMITH    02-DEC-81
 7934 MILLER    19 ACT    1 JAMES    22-JAN-82
 8156 MCMURRY   23 ACT    4 SMITH    18-APR-12
 8157 WALKER    24 ACT    4 SMITH    18-APR-12

23 rows selected.
```

In the EMP_ALL query above, EMPNO 7654 MARTIN has 2 instances of employment. The first instance was last modified/entered on April 18th, 1981 and deleted on May 14th, 1981 as captured by EMP ID 7. Later, MARTIN returned to the company, maintained the same EMPNO, but received a new instance of employment as captured by EMP ID 14. The STAT data shows that EMP ID is "ACT" or active, so MARTIN is still employed on this, the second instance of employment. In the example below, SMITH will be retired from the company by deleting the current employment instance from EMP_ACT.

```
SQL>
SQL> execute util.set_usr('MILLER');

PL/SQL procedure successfully completed.

SQL>
SQL> -- SMITH retires today
SQL> delete from emp_act
  2   where empno = 7369;

1 row deleted.

SQL>
SQL> select empno, ename, id eid, stat, dept_id did,
  2         aud_beg_usr, aud_beg_dtm, aud_end_usr, aud_end_dtm
  3   from emp_all order by empno, id;

EMPNO ENAME    EID STAT DID AUD_BEG_ AUD_BEG_D AUD_END_ AUD_END_D
----- -------- --- ---- --- -------- --------- -------- ---------
 7301 ELLISON    1 HIST   1 DAVIS    30-OCT-80 THOMPSON 30-JUN-81
 7344 DAVIS      2 HIST   1 SMITH    28-NOV-81 SMITH    10-DEC-81
 7369 SMITH      3 HIST   2 SMITH    28-FEB-83 MILLER   18-APR-12
 7499 ALLEN      4 ACT    3 THOMPSON 14-MAY-81
 7521 WARD       5 ACT    3 THOMPSON 14-MAY-81
 7566 JONES      6 ACT    2 SMITH    30-NOV-81
 7654 MARTIN     7 HIST   3 THOMPSON 18-APR-81 THOMPSON 14-MAY-81
 7654 MARTIN    14 ACT    3 SMITH    26-SEP-81
 7698 BLAKE      8 ACT    3 SMITH    28-NOV-81
 7782 CLARK      9 ACT    1 SMITH    01-DEC-81
 7788 SCOTT     10 HIST   2 THOMPSON 10-JUN-81 JAMES    07-MAR-82
 7788 SCOTT     20 ACT    4 SMITH    18-APR-12
 7839 KING      11 HIST   1 THOMPSON 14-JUN-81 SMITH    29-AUG-81
 7839 KING      15 ACT    1 SMITH    15-NOV-81
 7840 LANE      12 HIST   1 THOMPSON 15-AUG-81 SMITH    28-NOV-81
 7844 TURNER    13 ACT    3 SMITH    09-SEP-81
 7876 ADAMS     16 HIST   2 SMITH    24-NOV-81 JAMES    15-JUN-82
 7876 ADAMS     21 ACT    2 SMITH    09-JAN-83
 7900 JAMES     17 ACT    4 MILLER   18-APR-12
 7902 FORD      18 ACT    2 SMITH    02-DEC-81
 7934 MILLER    19 ACT    1 JAMES    22-JAN-82
 8156 MCMURRY   23 ACT    4 SMITH    18-APR-12
 8157 WALKER    24 ACT    4 SMITH    18-APR-12

23 rows selected.

SQL>
SQL> select empno, ename, id eid, dept_id did, aud_beg_usr, aud_beg_dtm
  2    from emp_act where empno = 7369;

no rows selected

SQL>
SQL> select empno, ename, emp_id eid, last_active, dept_id did,
  2        aud_beg_usr, aud_beg_dtm, aud_end_usr, aud_end_dtm
  3    from emp_hist where empno = 7369 order by aud_beg_dtm;

EMPNO ENAME    EID LAST MGR_EMP_ID DID AUD_BEG_ AUD_BEG_D AUD_END_ AUD_END_D
----- -------- --- ---- ---------- --- -------- --------- -------- ---------
 7369 THOMPSON   3               1   1 DAVIS    15-DEC-80 THOMPSON 25-JUN-81
 7369 THOMPSON   3              11   1 THOMPSON 25-JUN-81 SMITH    21-AUG-81
 7369 SMITH      3              12   1 SMITH    21-AUG-81 SMITH    01-DEC-81
 7369 SMITH      3              15   1 SMITH    01-DEC-81 SMITH    26-FEB-83
 7369 SMITH      3 Y            18   2 SMITH    26-FEB-83 MILLER   18-APR-12

5 rows selected.

SQL>
SQL> commit;

Commit complete.
```

Notice that the STAT of SMITH (EMP ID 3) has changed from ACT to HIST. This shows that EMP ID 3 is no longer an active instance of employment (no longer in EMP_ACT), which is confirmed by the EMP_ACT query. The last query shows that only 1 record, the last record, has a "Y" for LAST_ACTIVE. When the delete was performed on EMP_ACT, this flag was set to recognize there are no more active view records for this entity. Though EMP_ACT has many records for EID 3, the EMP_ALL view only returns the last active record.