ILP PROGRAM - ORACLE APPLICATIONS

Tata Consultancy Services

Oracle Forms Study Guide – Day1

Author: MD. Nazmul Hoda (TCS)

Creation Date: Jan 15, 2015
Last Updated: Jan 15, 2015
Document Ref: ILP/ORACLEAPPS/FORMS/01
Version: DRAFT 1A

Approvals:

<Approver 1>	Santanu Sarkar (TCS)
<Approver 2>	Shubho Chakraborty(TCS)

Contents

Table of Contents

Document Control

Change Record

Date	Author	Version	Change Reference
15-Jan-15	MD. Nazmul Hoda	Draft 1a	No Previous Document

Reviewers

Name	Position

Distribution

Copy No.	Name	Location
1	Library Master	Project Library
2		Project Manager
3		
4		

Note to Holders:

If you receive an electronic copy of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a hard copy of this document, please write your name on the front cover, for document control purposes.

How to use this manual

 Video1: Script: Vid1-Introduction to the chapter and its content

This video will introduce the material covered in this document which includes the following:

1. How this document is organized
2. What is the purpose of this document
3. What will you achieve after going through the document and related videos
4. How to read this document
5. How does it relate to the work you will be doing on real project
6. Reference to other reading materials for further references

---

This manual has been organized as a step by step guide to teach how to create reports using Oracle Developer Suite 10G. The target audience is new comes to Oracle Developer suite. It assumes that the reader has basic knowledge of Oracle concepts and PL/SQL. After completing this course, you will be able to crate variety of reports using Oracle Developer Suite 10G.

This manual is organized to be read in a serial fashion and follow the instructions given in the document as it is. Practical examples are given in each section to guide you through every step. The tables referred here are common (shared) tables used by different batches, so care should be taken not to delete or update the rows which does not belong to you, this may create problem for the other batches. At the end of the course, you should delete the data you have created.

There are several symbols used to designate particular sections, which are described below:

 - Describes the purpose of the section.

 - Notes relevent to the scetion above

 - This denotes the task to be completed by the audience on his own PC. The layout of the output has to be followed as it is. For any confusion, the faculty should be contacted.

1. Introduction of Forms

1.1 What is Forms and D2K

The **Internet Computing Solutions:** Oracle offers a range of tools and deployment options for Internet computing. Different types of developers and applications require different toolsets. Enterprise application developers need a declarative model-based approach. Oracle Designer and Oracle Forms Developer provide this solution, as do the later versions of Developer. Focuses on how you can use Oracle Forms Developer to rapidly build scalable, high-performance applications for the Internet and then deploy the applications with Oracle Forms Services.

Application Type and Audience	Product Approach	Oracle Products
Enterprise applications, Business developers	*Repository-based modeling & generation, Declarative*	*Oracle Designer, Oracle Forms Developer, & Oracle Forms Services*
Java components, Component developers	Two-way coding, Java and JavaBeans	Oracle JDeveloper Oracle Application Server 10g
Self-service applications & content management, Web site developers	Browser-based, Dynamic HTML	Oracle Portal Oracle Database Server
Reporting and analytical applications, MIS & business users	Dynamic Web reporting, Drill, Analyzing, Forecasting	Oracle Reports Developer, Oracle Reports Services, Oracle Discoverer, & Oracle Express

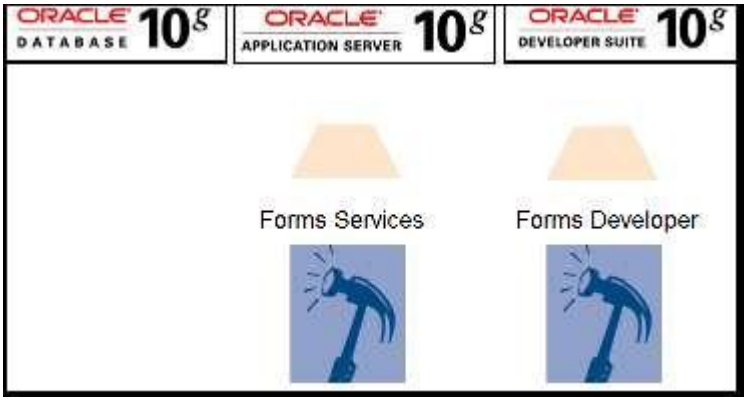
For Web site developers and content publishers who want to build self-service dynamic Hypertext Markup Language (HTML) applications for Web sites, Oracle Portal provides an easy-to-use development environment that resides entirely inside an Oracle database. Portal provides a browser-based environment from development through deployment of an application.

For Management Information System (MIS) developers and end users, there is the Oracle Business Intelligence toolset. Oracle Reports Developer, Oracle Reports Services, Oracle Discoverer, and Oracle Express provide the whole range for reporting, analysis, and trending facilities

1.2 Why it's used:

The ***Oracle Database*** manages all of your information, such as Words like documents, Data like excel spreadsheets, XML Data, and even images. Oracle tools such as Forms can automatically reuse the database structure and its integrity constraints, which reduces the amount of manual coding.

The ***Oracle Application Server*** Runs all of your applications, including Java, wireless, portals, and business intelligence. Using Oracle Application Server, you can deploy and manage in a single application server all applications developed with Oracle Developer Suite. ***Oracle Application Server contains*** Oracle Forms Services, which you use to deploy your Forms applications.



Oracle Developer Suite: Leverages the infrastructure offered by Oracle Application Server and Oracle Database, enabling developers to quickly and easily build scalable, secure, and reliable e-business applications. The suite provides a complete and highly productive development environment for building applications. Oracle Forms Developer, which you use to build Forms applications, is part of Oracle Developer Suite.

As a purpose of only simple introduction, The Oracle Application Server has a layered architecture including the following services:

Communication Services: Communication management for a variety of protocols

Application Services: J2EE Container that provides a common run-time environment for applications developed as JSPs, Servlets,

EJBs, and Web Services

System Services: A common set of run-time services that are necessary for J2EE **Applications and Web Services**, such as request dispatch and scheduling, resource management, resource pooling, clustering, fault monitoring, transaction management, and messaging

Management Services: A common set of systems management services to monitor the status, performance, and faults of the system; to monitor resource consumption and usage; to manage a single instance or cluster of instances; to centrally administer security for users and applications; and to provide a comprehensive directory service framework to manage users

Connectivity Services: Provide connectivity to a variety of systems

Solutions: A comprehensive set of solutions, all built on the infrastructure described above including Enterprise Portals, Enterprise Integration, Business Intelligence, Wireless, and ISV Solutions

Integration and Commerce Services: Provide ability to build enterprise portals, integrate systems with each other, automate business processes, and serve personalized recommendations to users

1.3 Form with OC4J - Overview

- What Is OC4J?
- Oracle Application Server Containers for J2EE (OC4J) is Oracle's Java 2 Enterprise Edition (J2EE) container that executes on any Java Virtual Machine (JVM), which is the Java interpreter that is provided on each

operating system and hardware platform. OC4J is implemented completely in Java, making it lightweight and easy to install. At the same time, it provides complete support for J2EE applications, including servlets, Enterprise JavaBeans, Java Server Pages, and so on.

- OC4J is ideally suited to run Forms applications. It is included in Oracle Developer Suite to enable you to test your applications, if desired, on the same machine where you are running Forms Builder. That is, you do not need to install Oracle Application Server to test your applications.
- Oracle Application Server Containers for J2EE (OC4J) is:
- Preferred to run Forms applications
- Included with Oracle Developer Suite to enable testing

1.4 Form: Starting OC4J

To use OC4J on Windows, start it by executing the batch file provided, called startinst.bat. This file is located in the j2ee\DevSuite\ subdirectory of the Developer Suite home directory. If you want to test your applications on your client machine later, it is a good idea to set up a shortcut to this batch file, and also to the batch file called stopinst.bat, which stops the OC4J instance. Alternatively, you can call these batch files from the

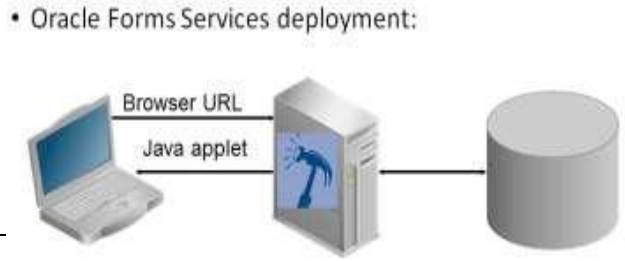
Windows Start menu: Programs > Oracle Developer Suite > Forms Developer > Start [Shutdown] OC4J Instance.

The batch file executes in a separate window, which you can minimize if desired. Do not close this window, or you will abort the OC4J instance.

When you no longer need to run OC4J, you can execute the batch file called stopinst.bat to stop the OC4J instance.

1.5 Running a Form

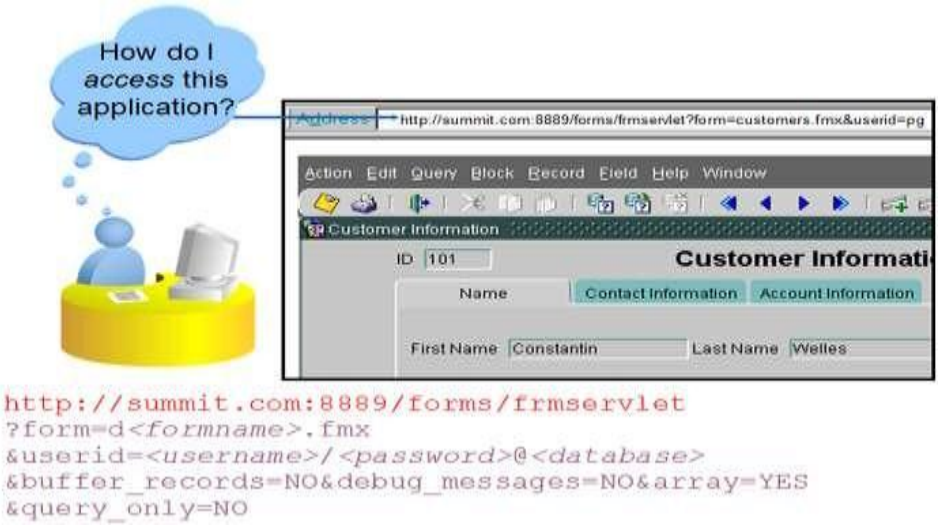
Deploying Forms applications to the Web is implemented by the three-tier architecture of Oracle Application Server. Application logic and the Forms Runtime Engine reside on the middle-tier application server. All trigger processing occurs on database and application processing occurs servers, whereas user interface run Forms on the Forms Client. End users can browser. Developer applications in a Web



Users request an application in their Web browsers by entering a URL that points to the application. Forms Services then generates an HTML file that downloads a Java applet to the client machine. This small applet is capable of displaying the user interface of any form, while the application logic is executed on the middle tier.

1.6 Running a Form: Browser

The URL to invoke an application must have the following format:

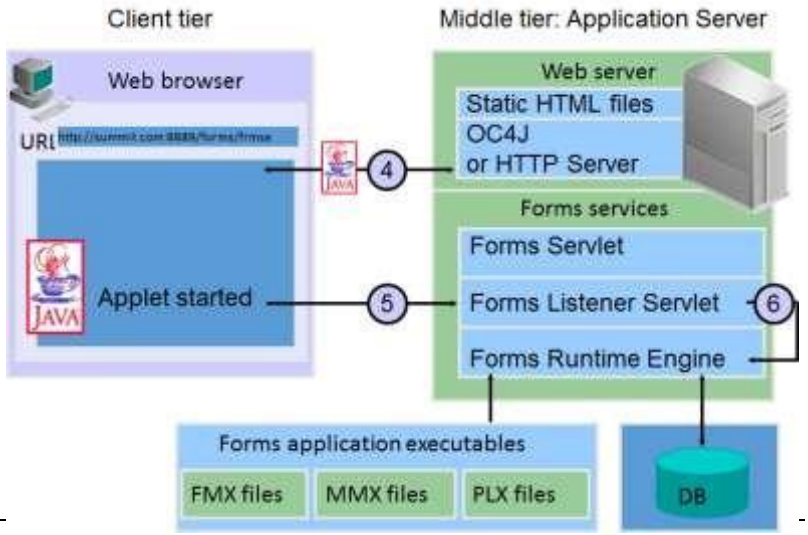


http://host.domain[:port]/forms servlet or html file [parameters] (optional portions of URL enclosed in brackets)

1.7 Starting a Run-Time Session

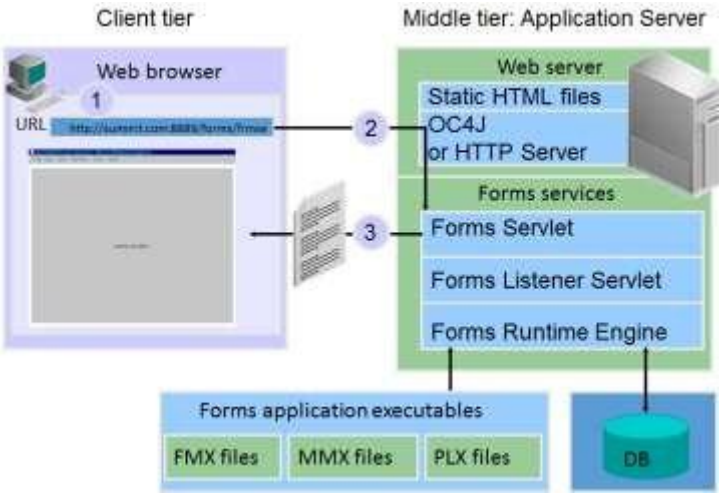
Starting a run-time session involves the following steps:

1. The user accesses the URL that indicates that a Forms application should be run.
2. The Oracle HTTP Server or OC4J receives an HTTP request from the browser client and contacts the Forms Servlet.
3. The Forms Servlet dynamically creates an HTML page containing all the information to start the Forms session.



4. The Oracle HTTP Server or OC4J downloads a generic applet to the client after checking that it has not already been downloaded. The client caches the applet so that it can run future Forms applications without downloading it again.

5. The client applet contacts the Forms Listener Servlet to start the session. The Forms Listener Servlet starts an instance of the Forms Runtime Engine on the Forms Server (middle tier). If included in the HTML file, Forms Runtime command-line parameters (such as form name, user ID and password, database SID, and so on) and any user-defined Forms Builder parameters are passed to the process by the Forms Listener Servlet.



6. The Forms Listener Servlet establishes a connection with the Forms Runtime Engine, which connects to the database if needed and loads application executable files.

7. The Forms applet displays the user interface of the application in the main window of the user's Web browser.

8. The Forms Listener Servlet, working through OC4J or the HTTP Server, manages communication between the Forms applet and the Forms Runtime Engine.

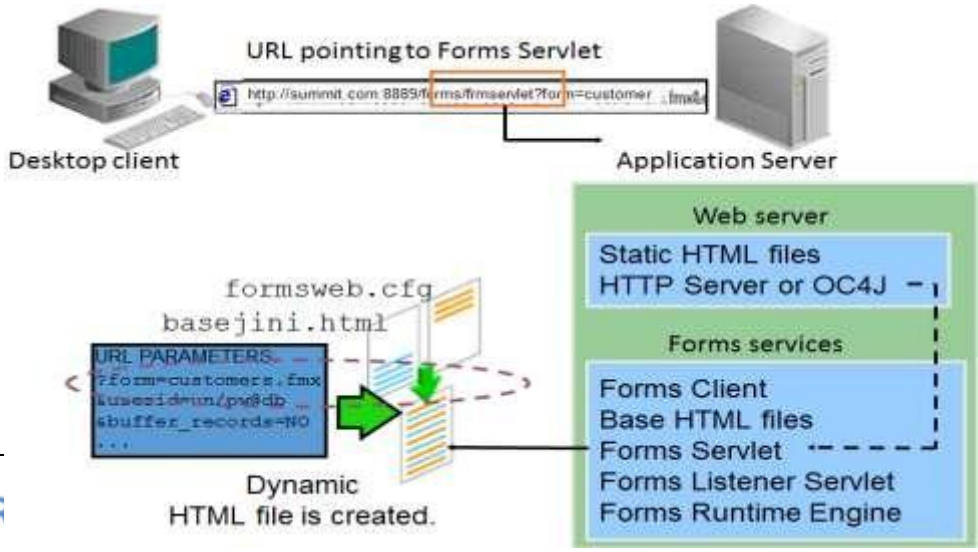
1.8 Forms Servlet

The Forms Servlet is a Java servlet that creates a dynamic HTML file by merging information from the following sources:

- The Forms Web configuration file. The Forms base HTML file
- The application's URL parameters

1.9 Forms Developer Executable

- Forms Builder includes the following two executables (components) that you can access as the designer of applications.
- **Forms Builder:** This is the application-building



component of Oracle Forms Developer. You can use Forms Builder to design and store the definitions of form, menu, and library documents. While in the Forms Builder, you can invoke the other component, Forms Compiler. You must run the Forms Builder component in a GUI environment in order to use its graphical design facilities.

- **Forms Compiler:** After your form is built, use the Forms Compiler. This reads the definition of your module and creates an executable run file.
- **Invoking Forms Builder Executables**
- In a GUI environment, you usually store commands to invoke Forms Builder components in menus and icons for convenient access. You can also enter these commands on the command line.

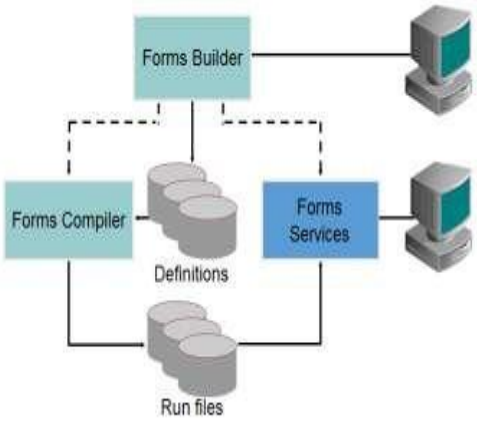
For example: FRMBLD [my_form] [scott/tiger@my_database]

1.10 Forms Services

Because Forms applications are Web based, it is not possible to run them directly from the command line. Instead, they are invoked by entering a URL, directed to Forms Services, in a browser.

The files used at run time must already have been compiled by the Forms Compiler component. These files must reside on the middle-tier machine in a directory accessible to the Forms Runtime Engine (in FORMS_PATH).

To test your applications, you can also access Forms Services directly from Forms Builder by setting certain preferences, as described later in this lesson.



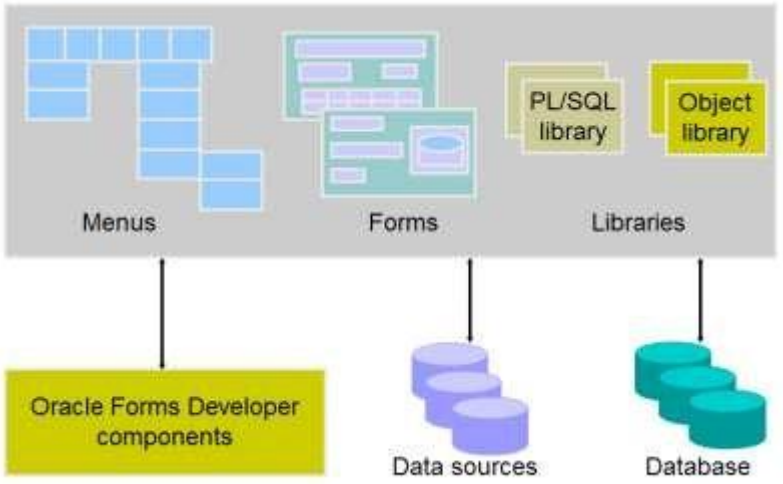
1.11 Form Modules

A Forms application can consist of many modules—that is, files. A module is a major component of your application and is the basis for storage and ownership. A module owns the objects that it contains.

A Forms Developer module can be of the following types:

1. **Form:** As the main component of an application, the form module presents the objects and data that users can see or interact with. Data items in a form are arranged into records.

2. **Menu:** A menu module can consist of a hierarchy of menus, each with selectable items. Forms Builder provides the default menu for every form. The default menu



includes commands for all basic database operations, such as insert, delete, query, and so on. If your application has specific requirements that are not met by the default menu, then you can create a custom menu module. You can use a menu module with multiple forms.

3. **PL/SQL Library:** A PL/SQL library is a collection of PL/SQL program units whose code can be referenced and called from other modules. PL/SQL library documents can contain program units that can be used by other form and menu modules.
4. **Object Library:** An object library is a collection of form objects that you can use in other modules. You can create it to store, maintain, and distribute standard objects that can be reused across the entire development organization.

You can build an application from multiple form modules, menu modules, and library documents as needed. A form module can be run independently, but menu modules, PL/SQL libraries, and object libraries are functional only when attached to or included in a form module.

Forms Builder: Key Features

Forms Builder is a major component of Oracle Forms Developer. You can use Forms Builder to quickly develop form-based applications for presenting and manipulating data in a variety of ways.

- Users of Forms Builder applications can:
- Insert, update, delete, and query data by using a variety of interface items
- Present data by using text, image, and custom controls, including JavaBeans and Pluggable Java Components
- Control forms across several windows and database transactions
- Access comprehensive facilities by using integrated menus
- Send data directly to Oracle Reports
- As the designer of Forms Builder applications, you can:
- Design forms that use a number of data sources, including Oracle databases
- Build applications quickly and easily by using powerful GUI development tools
- Design applications for Internet deployment
- Copy and move objects and their properties easily between applications
- Use design features such as wizards, the Layout Editor, Object Navigator, and PL/SQL Editor

With Forms Builder, you can:

- Provide an interface for users to insert, update, delete, and query data
- Present data as text, image, and custom controls
- Control forms across several windows and database transactions
- Use integrated menus
- Send data to Oracle Reports

Video1: Script: Introduction to different features and components of form

This video will give an explanation of the sections discussed above with practical examples, which will include: (This is going to be mainly a face recording, and small screen captures padded – like A running form, a running OC4J):

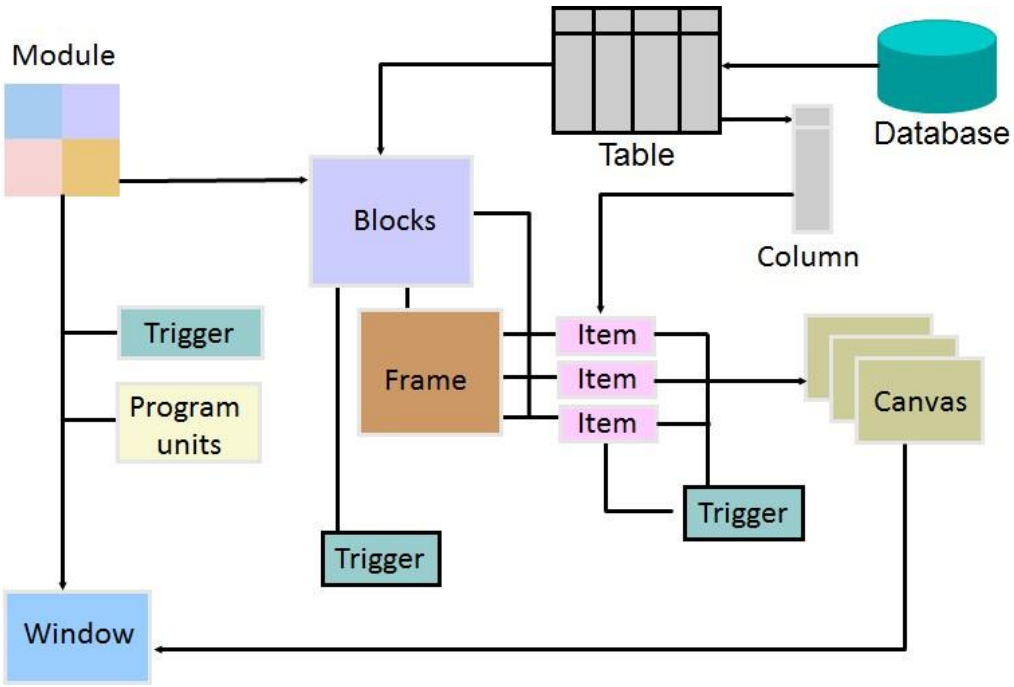
1. Why is the use of a form, show some heavily used seeded forms and some custom forms
2. Walkthrough the Forms Components on a High level
3. Brief introduction to OC4j, and its relationship with Forms, How to start and stop OC4j services, errors in its absence.
4. Demonstrate, how to run a Form
5. Walkthrough the File types used in forms
6. Walkthrough the Forms Services
7. Introduce the Form Modules - Menu, Library, layout editor etc.

Form 1.1 – Show a sample form with features and validations.
Forms 1.2 – Demonstrate a form from within form developer.

2. Building Blocks of Forms

2.1 Object Hierarchy

You can create many types of objects in a form module. They are discussed in more detail in later lessons. In the following table, note that some objects are associated, even though one might not be "owned" by the other.



2.2 Starting the Builder using Object Navigator

- The interface components of the Forms Builder tool help to provide the flexibility and productivity of the Oracle Forms Developer development environment.
- The Object Navigator is a hierarchical browsing and editing interface. You can use the Object Navigator to locate and manipulate application objects quickly and easily. Features include:
  - Client-side and server-side objects displayed hierarchically
  - Toolbar to create, delete or unload, and expand or contract
  - Icons to represent objects
  - Fast search feature

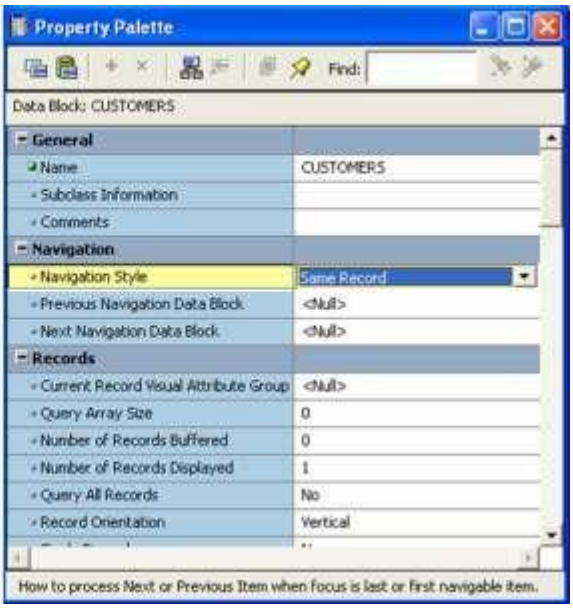
- A hierarchy represented by indentation and expandable nodes (Top-level nodes show module types, database objects, and built-in packages.
  All other nodes and the objects they contain are indented to indicate that they belong to these higher-level nodes.) Find field and icons, enabling forward and
  Backward searches for any level of node or for an individual item in a node

- Icons in the vertical toolbar to create, delete or unload, and expand or contract

2.3 Property Palette

All objects in a module, including the module itself, have properties that you can see and modify in the Property Palette. Features include:

- Copy and paste properties
- Fast search feature



Copy and reuse properties from another object Find field and icons, similar to the Object Navigator

Video2: Script: Introduction to Basic Building Blocks of form – covered in video 1.2 only.

This video will give an explanation of the sections discussed above with practical examples and screen captures, which will include:

1. Object Hierarchy - Organization of the building blocks of a form, walkthrough of the diagram and video captures to show different components in a real form builder and a running form.

2. Starting the Builder using Object Navigator - Video capture of how to start a form builder and introduction to its components.
3. Property Palette - Screen capture to show the property palette and significance of different fields

3. Getting Started in the Forms Builder.

To start Forms Builder, invoke it from the Windows Start menu: *Programs > Oracle Developer Suite > Forms Developer > Forms Builder.*

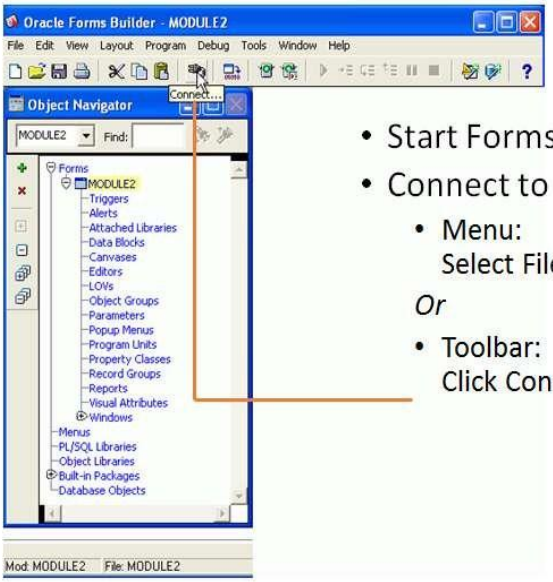
- Choose one of the following methods:
  - Use wizards:
    - Data Block Wizard
    - Layout Wizard
  - Build module manually.
  - Use the template form.

When you invoke Forms Builder, a Welcome dialog box appears. If you click Cancel to dismiss the dialog box, you see the Object Navigator and an empty new module.

If you build applications that access database objects, you must connect to a database account from the Forms Builder. Connect to a database if you need to:

Access database objects in the Object Navigator

Create Oracle Forms Developer objects that are based on database objects



- Start Forms Builder.
- Connect to the database:
  - Menu:
    Select File > Connect.

  Or
  - Toolbar:
    Click Connect.

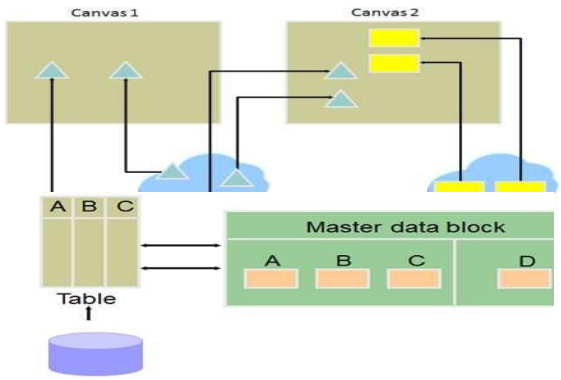
3.1 Forms Module Components

The three major objects in a form are:

Items: These are interface objects that present data values to the user or enable the user to interact with the form, depending upon the item type. There are several types of items. Items are logically grouped into blocks and visibly arranged on canvases.

Blocks: A block is the intermediate building unit for forms. Each form consists of one or more blocks. A block is the logical owner of items, and each item in a form belongs to a block. Items in one block are logically related *(for example, they may correspond to columns in the same database table or may need to be part of the same navigation cycle).*

Blocks, therefore, provide a mechanism for grouping related items into a functional unit for storing, displaying, and manipulating records.



Canvases: A canvas is a "surface" where visual objects, such as graphics and items, are arranged. A form module can have several canvases ***(such as the pages of a paper form).*** A canvas can display items from one or more blocks.

To see a canvas and its items, you must display the canvas in a window. By default, all canvases in a form appear in the same window (which could mean you see only one canvas at a time), but you can assign separate windows for each canvas so that several canvases can be viewed at once.

3.2 Types of Blocks:

1. Data Blocks:

When you build database applications with Forms Builder, many of the blocks will be data blocks. A data block is associated with a specific database table (or view), a stored procedure, a FROM clause query, or transactional triggers. If it is based on a table (or view), the data block can be based on only one base table, even though the data block can be programmed to access data from more than one table and data sources. By default, the association between a data block and the database enables the user to automatically access and manipulate data in the database.

However, to access data from other tables (non-base tables), you need to write triggers.

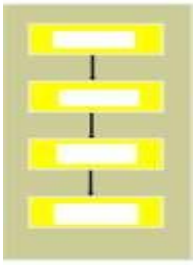
For a base table, Forms Builder can automatically perform the following actions:

- Create items in the data block to correspond to columns in the table. (These items are data items or base-table items.)
- Produce code in the form to employ the rules of the table's constraints.
- Generate SQL at run time (implicit SQL) to insert, update, delete, and query rows in the base table, based on the user's actions.

2. Control Blocks

A control block is not associated with a database, and its items do not relate to any columns

Within any database table. Its items are called control items. For example, you can create many buttons in your module to initiate certain actions, and you can logically group these buttons in a control block. The Single Form Control Module


Single Form Module

3. Master Versus Detail Blocks

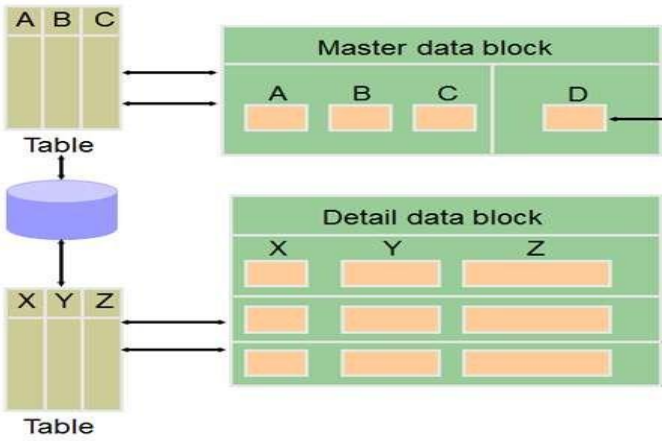
To support the relationship between data blocks and their underlying base tables, you can define one data block as the detail (child) of a master (parent) data block. This links primary key and foreign key values across data blocks, and synchronizes the data that these data blocks display.

Forms Builder automatically generates the objects and code needed to support master-detail relationships. As the designer, you need only request it.

Note: If your application requires it, you can also create independent data blocks in which there is no relationship between the two data blocks.

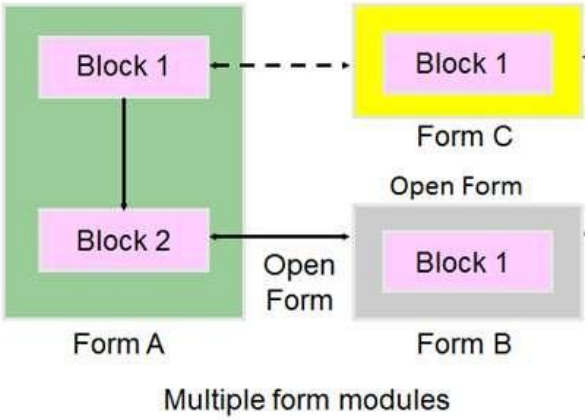
4. Single-Record Vs Multi-Record Blocks

You can design a data block to show one record at a time (single-record block) or several records at once (multi-record block). Usually, you create a single-record data block to show master block data and a multi record data block to show detail block data. In either case, records in a data block that are currently not visible on the screen are stored in a block buffer.



3.3 Multi-Form Applications

You can design a Form with multiple Data Block and Limited level of Control Block. Period of time later in the project to integrate all in certain roof level, the Parameter can function as a major role. The logical approach of Form integration as show.



Multiple form modules

3.4 Naming a New Form Module

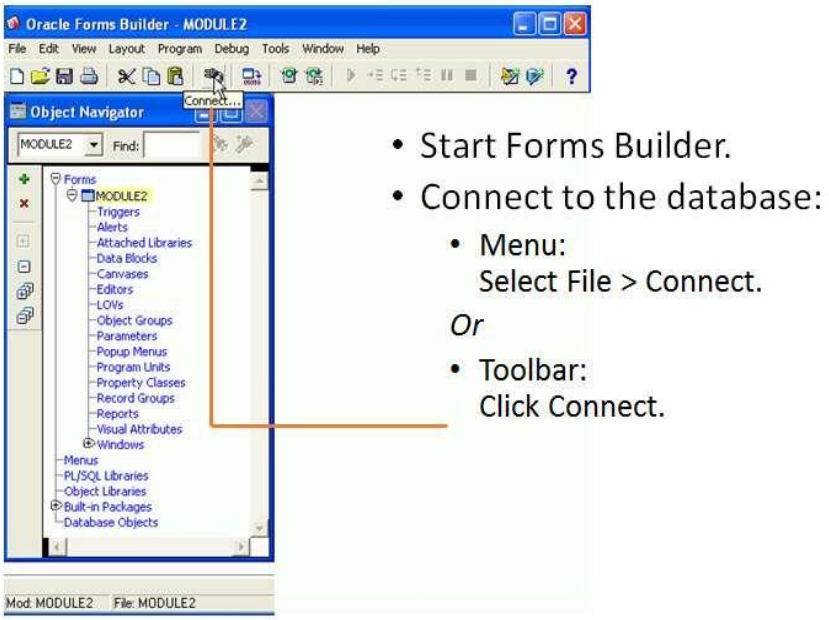
Each form module consists of several objects. Objects within a form, and the form module itself, have properties that define their behavior. You can see the properties of an object and their values in the Property Palette of the object

First to Create a Form as Double click on Programs > Oracle Developer Suite > Forms Developer > Forms Builder.

To open the Property Palette of an object, perform one of the following steps:

Double-click the object's icon in the Object Navigator.

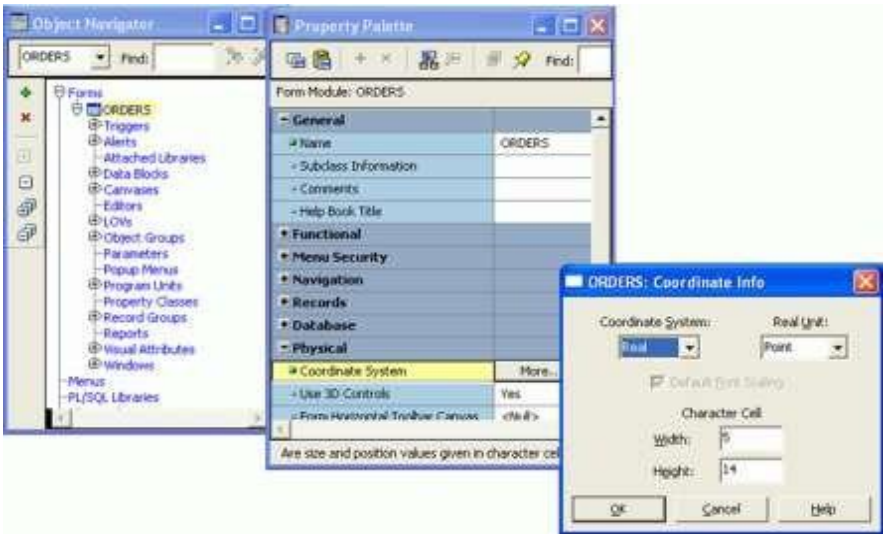
Double-click the object in the Layout Editor



Select the object in the Object Navigator and select Tools > Property Palette.

Right-click the object in the Object Navigator or in the Layout Editor and select Property Palette from the context menu.

To obtain online Help for any of the properties, click the property and use the Help key ([F1]), to display a description of that property.



You can define the properties of the form module when you first create it or modify the properties later. The properties affect the general behavior of the form and the objects within it.

Properties for a form module include the following:

Name

When you click Name Value and modify the default system generate Random Name. In the example the name has to change as ORDERS.

Coordinate System

When you click more in the Property Palette window with the Coordinate System property selected, the Coordinate Info window is displayed. The Coordinate System unit for a form can be one of the following:

Real: Unit can be pixel, centimeter, inch, point, or deci point.

Real units are suitable for GUI applications and enable flexibility and fine alignment when adjusting object positions and sizes.

Character: Units are character cells (default size taken from the default font settings).

The default unit is point (Real). This means that object positions and sizes within the form are measured by this unit. Points provide fine alignment and consistency across different platforms and video devices.

Video3: Script: Getting Started with Form Builder

This video will give an explanation of the sections discussed above with practical examples and screen captures, which will include: Getting Started in the Forms Builder - Screenplay of Canvas, Blocks, Data Block, Control Block, Master-Detail Blocks, and Single/Multi Record Block.

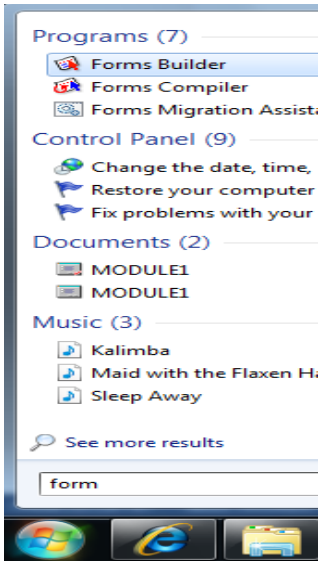
4. Creating a Form Module

 In this section you will learn how to create a simple form

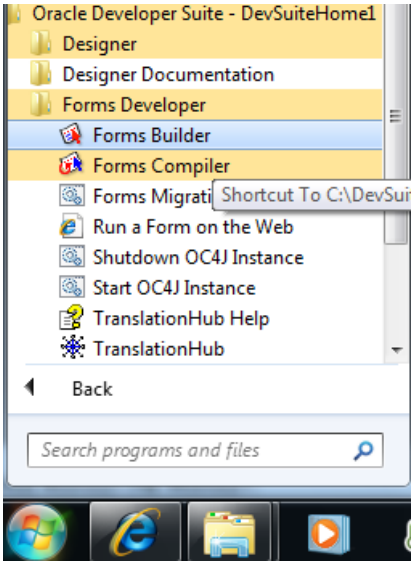
Before creating the form, we will create the table and data. Check if the table exists, else create and populate the table using following script:

```
CREATE TABLE MATERIAL_MASTER (
MATERIAL_ID NUMBER,
MATERIAL_NAME VARCHAR2(200),
MATERIAL_TYPE VARCHAR2(20),
MATERIAL_UOM VARCHAR2(10),
UOM_SIZE  NUMBER,
BOM_ID NUMBER)
/
    INSERT INTO MATERIAL_MASTER VALUES (100, 'A4 PAPER','STATIONERY','BUNDLE',100,10)
    /
    INSERT INTO MATERIAL_MASTER VALUES (101, 'BALL PEN','STATIONERY','BUNDLE',10,11)
    /
    INSERT INTO MATERIAL_MASTER VALUES (102, 'SHREDDER','STATIONERY','NUMBERS',1,12)
    /
    INSERT INTO MATERIAL_MASTER VALUES (103, 'BEETEL PHONE
    HANDSET','ELECTRONICS','NUMBER',1,13)
    /
    INSERT INTO MATERIAL_MASTER VALUES (104, 'ARMCHAIR WITH
    HANDREST','FURNITURE','NUMBER',1,14)
    /
    INSERT INTO MATERIAL_MASTER VALUES (105, 'HANDWASH','TOILETRIES','LITRE',20,15);
    /
```

Now we will create the form. Open the form builder – You can either do it by Start > Run, type form, the icon for Forms Builder, or Start> All Programs> Oracle Developer Suite – DevSuittHome1 > Forms Builder



Start > Run, type form

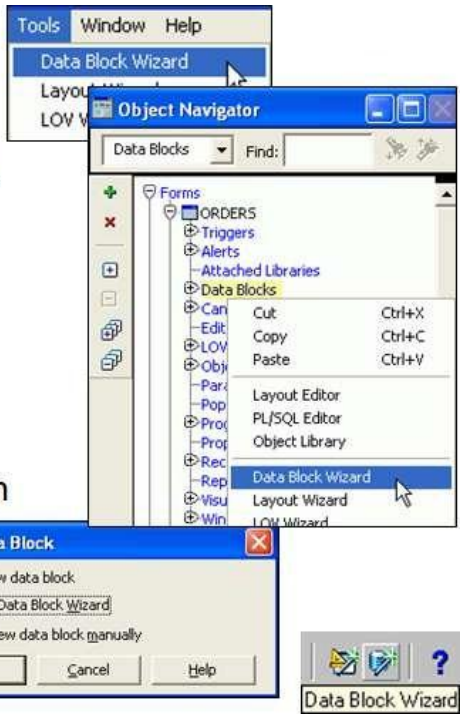


Start> All Programs> Oracle Developer Suite – DevSuittHome1 > Forms Builder

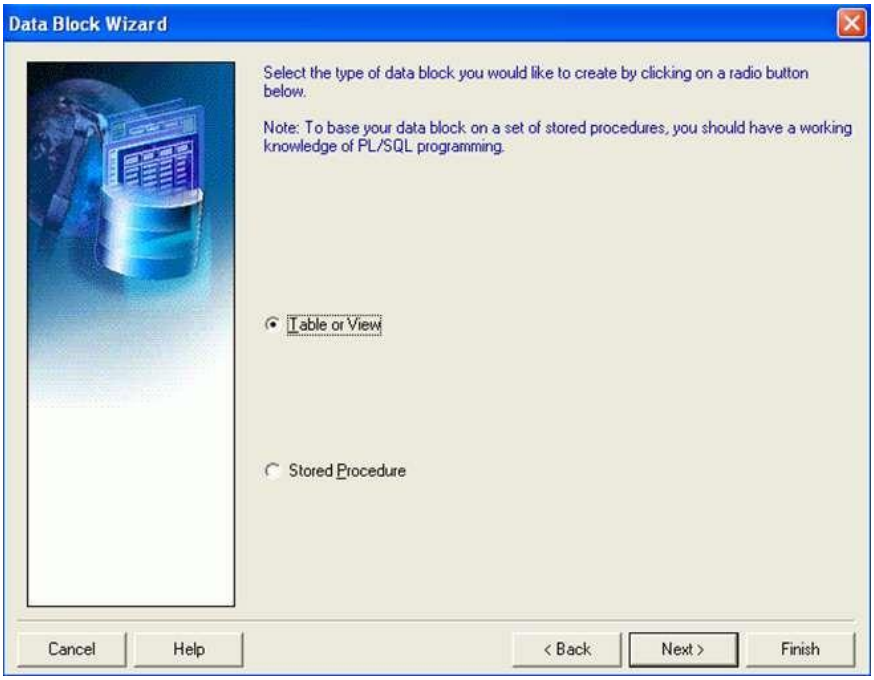
Upon launching the Form Builder, invoke the Data Block Wizard:

Select Tools > Data Block Wizard from the Forms Builder default menu system. Right-click and select the Data Block Wizard option. In the Object Navigator, select the Data Blocks node, then click the Create icon. In the New Data Block dialog box, select the Use the Data Block Wizard option. Click Data Block Wizard on the toolbar.

- In Forms Builder, do one of the following:
  - Select Tools > Data Block Wizard.
  - Right-click and select Data Block Wizard.
  - Select the Data Blocks node and click Create icon; select Use the Data Block Wizard option.
  - Use the Data Block Wizard button on the toolbar.



Select “Use the Data Block Wizard” option and follows the Steps

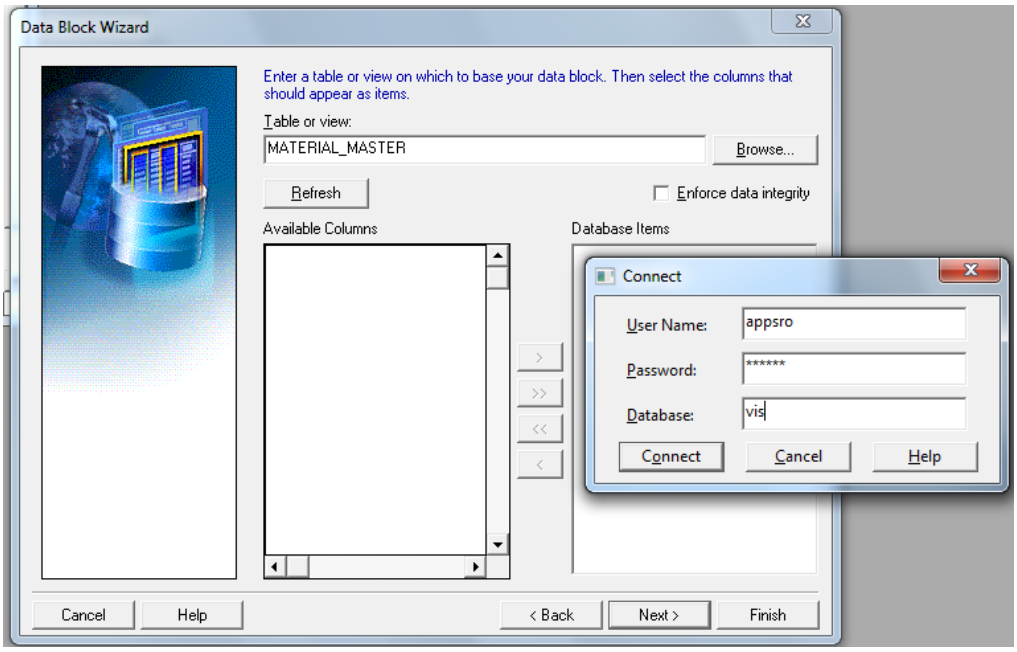


Step 1: Choose between one of two data source types:

- Table or View
- Stored Procedure

Select the Table or View (default) option.

Refer the section for detailed full Data Block Creation



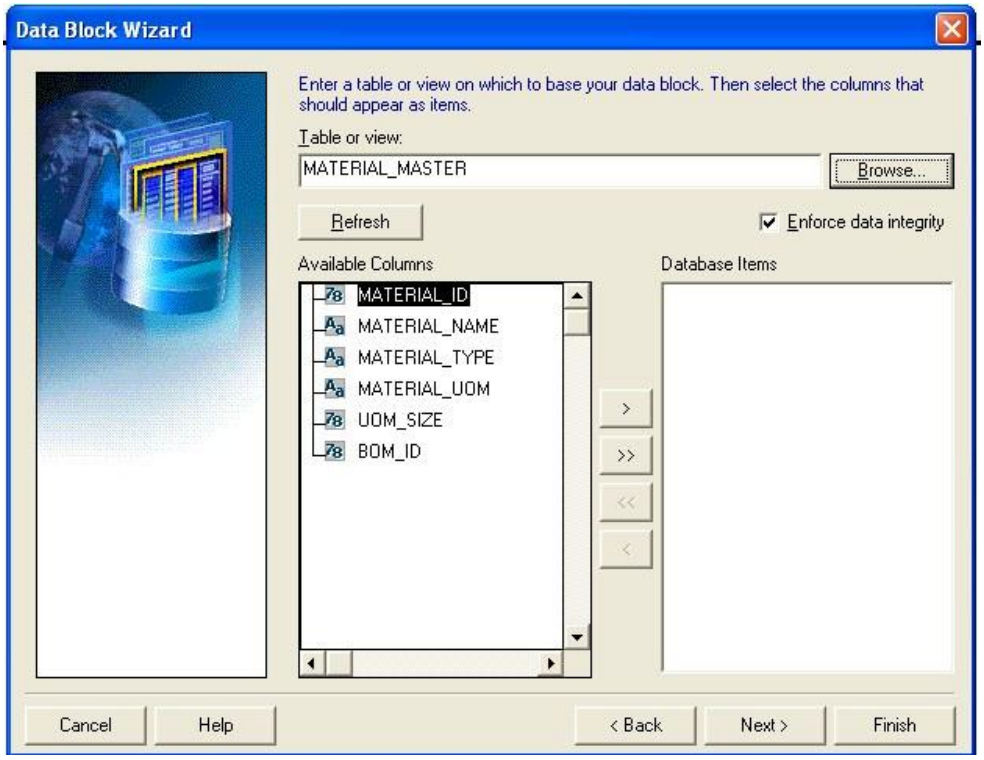
Step 2: Table Page

Enter the table or view name for the data source name, or click Browse and select a name from a dialog box. Click Refresh to display a list of columns in the selected table or view. If you are not connected to the database, the Connect box is displayed.

Select the columns you want to include in the data block. (To select more than one column, press and hold [Ctrl] and then select the columns.)

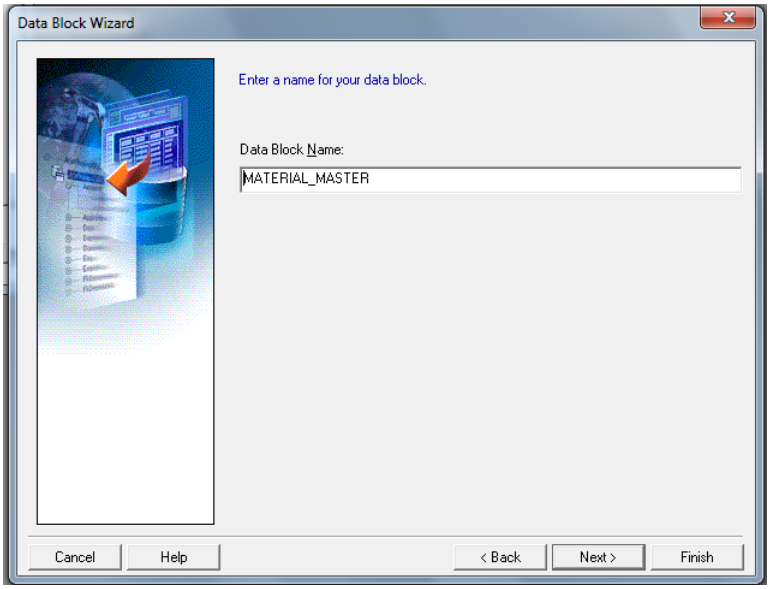
Click the double-right arrow or the double-left arrow to include or exclude all columns, or click the right arrow or the left arrow to include or exclude selected columns only.

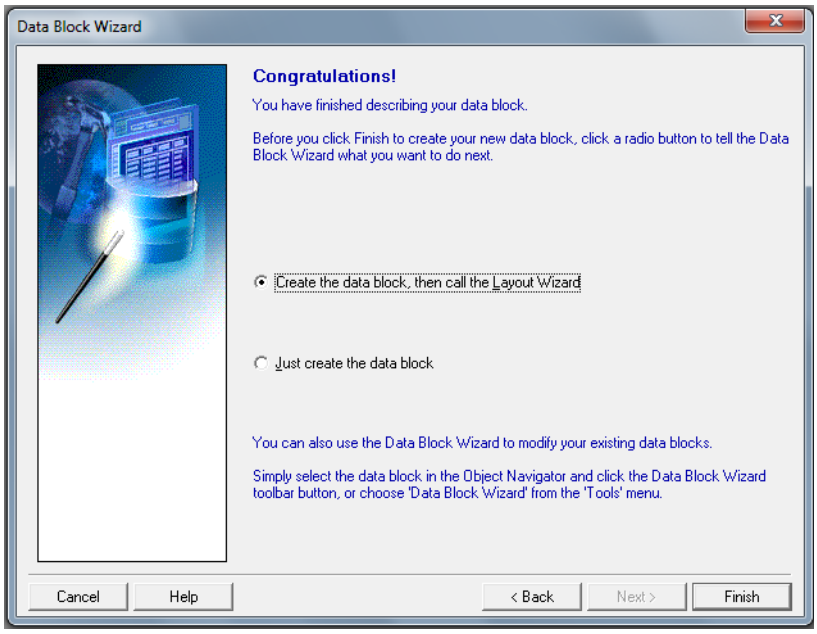
Select the “Enforce data integrity” check box if you want the wizard to enforce the database integrity constraints.



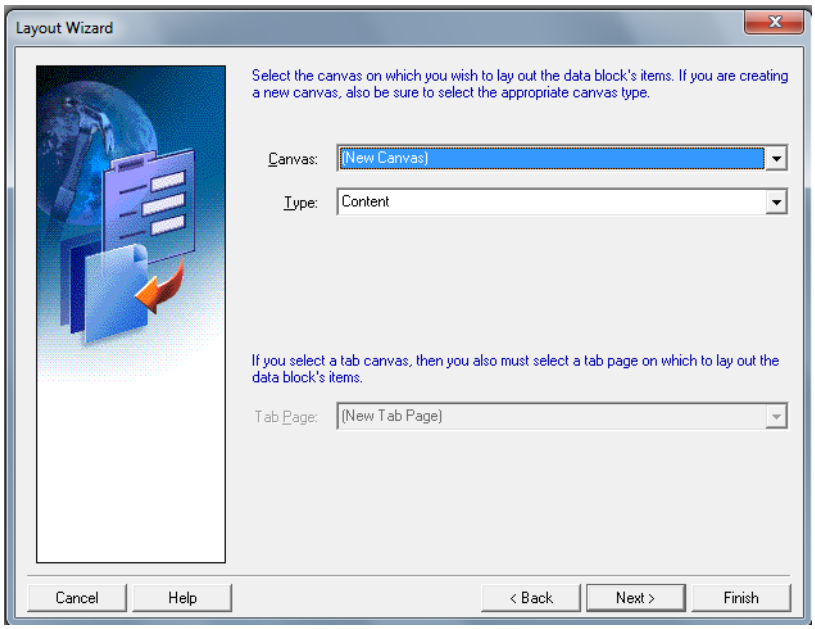
Note: If there is at least one other existing block in the current module, the next page that displays is the Master-Detail page, where you can associate the new data block with other master data blocks. In this case, we will select all the fields by pressing the double arrow button, and click Next.

This will show the Data Block Name as the table name by default, with an option to change it, we will accept the default.

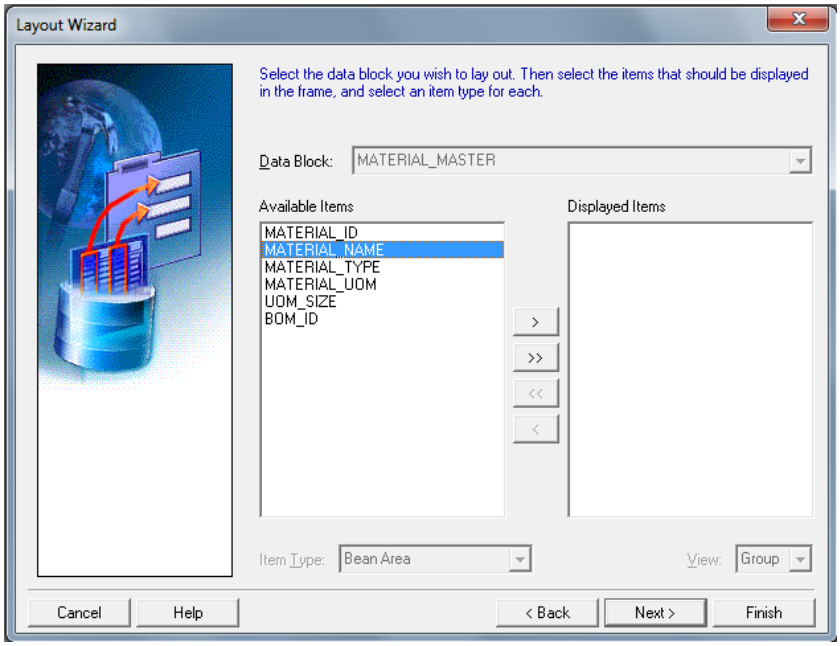




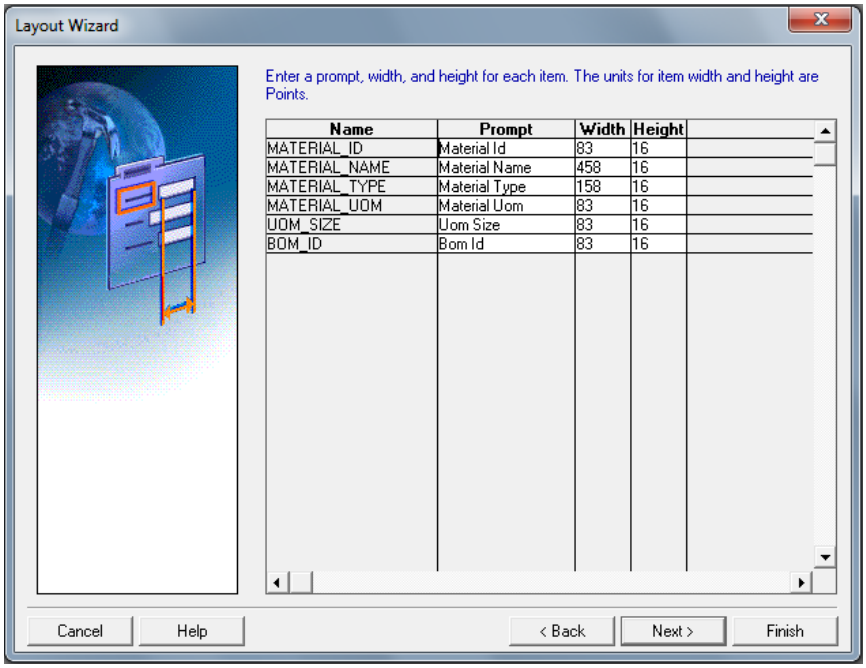
Select the first option as above and Click Finish. This will pop the Layout Wizard.



Click Next

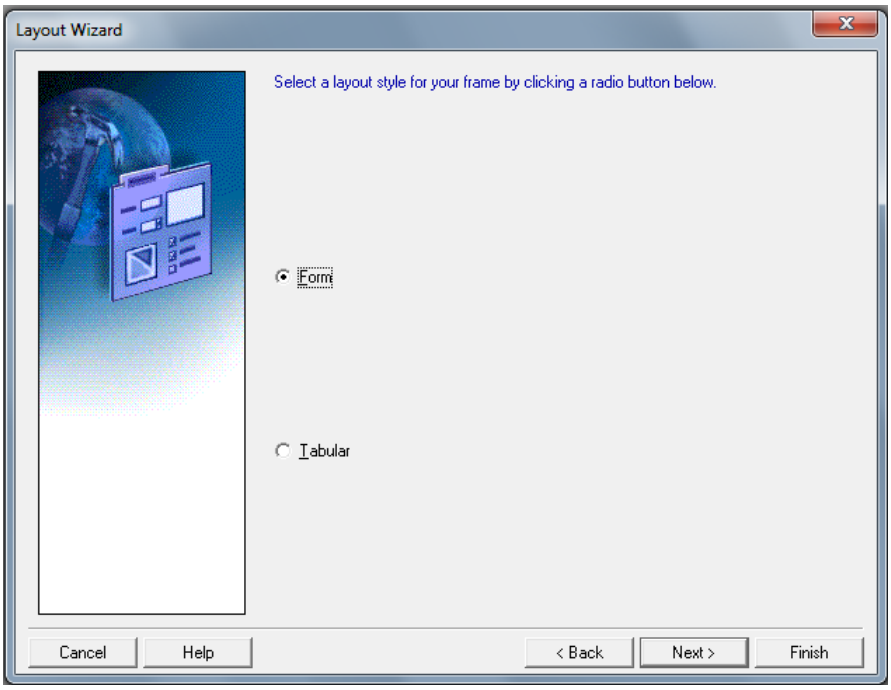


Select the fields to be displayed in the form, we will select by clicking the double arrow, and click Next. Now you have the option of changing the prompt name and the labels. You can change it, or leave it as is it, you will get the chance to change this later on also.

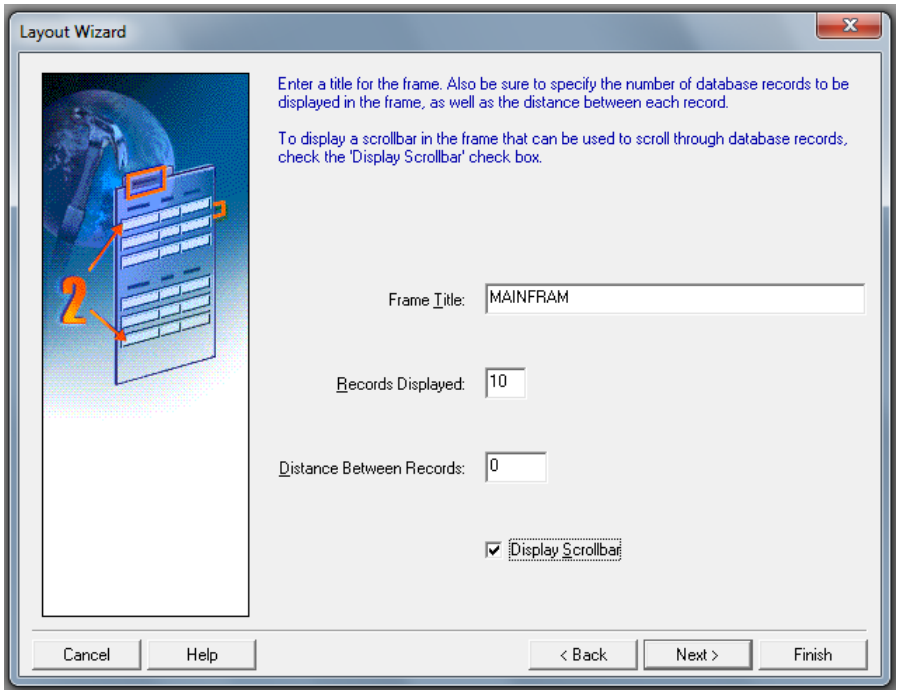


Select the layout style.

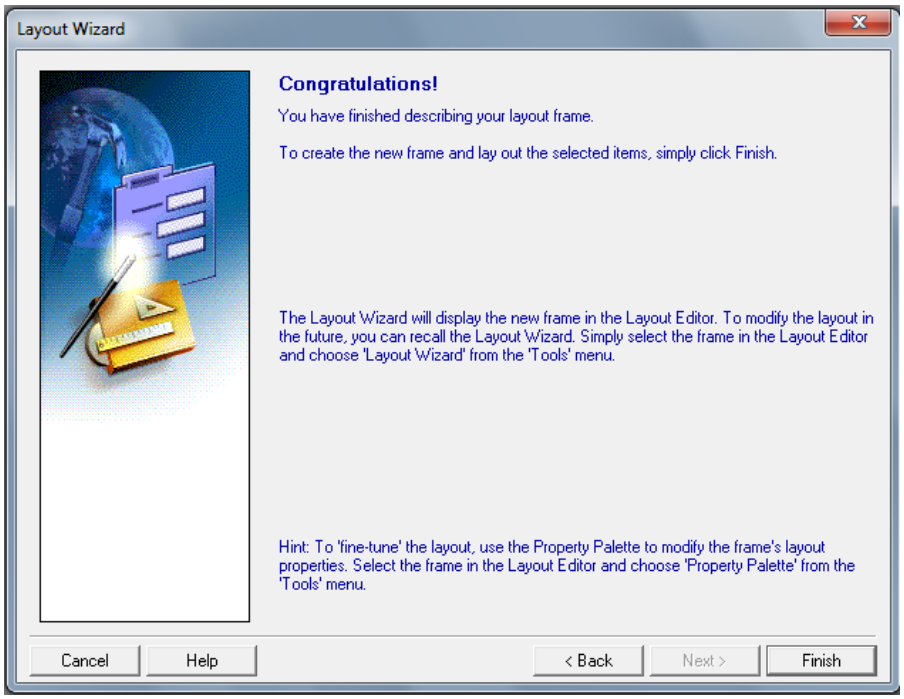
Form layout means one data record will be displayed in at a time, and fields will be scattered across the page. Tabular Layout means one data record will be displayed on each line horizontally, and multiple data records will be displayed in one page. We will select the Tabular layout.



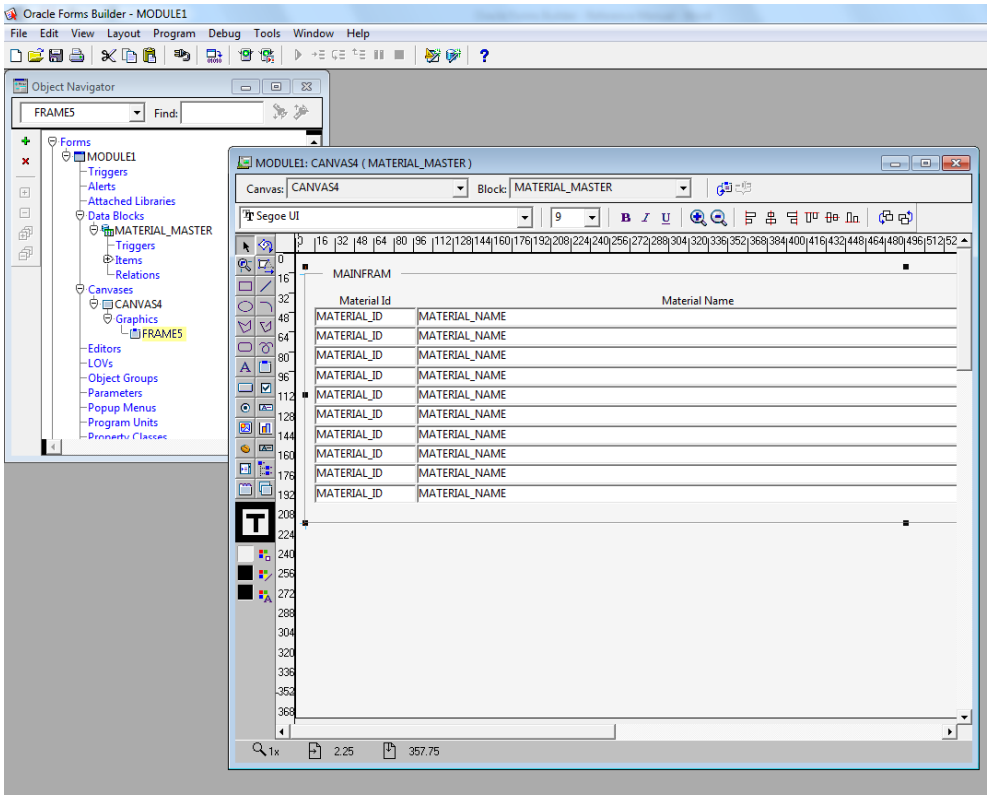
Since we have selected Tabular type, it will ask for the number of records to be displayed at a time, and if you want the scroll bar at the side to scroll through the records. We also need to provide the frame name as below:



We have created the form now

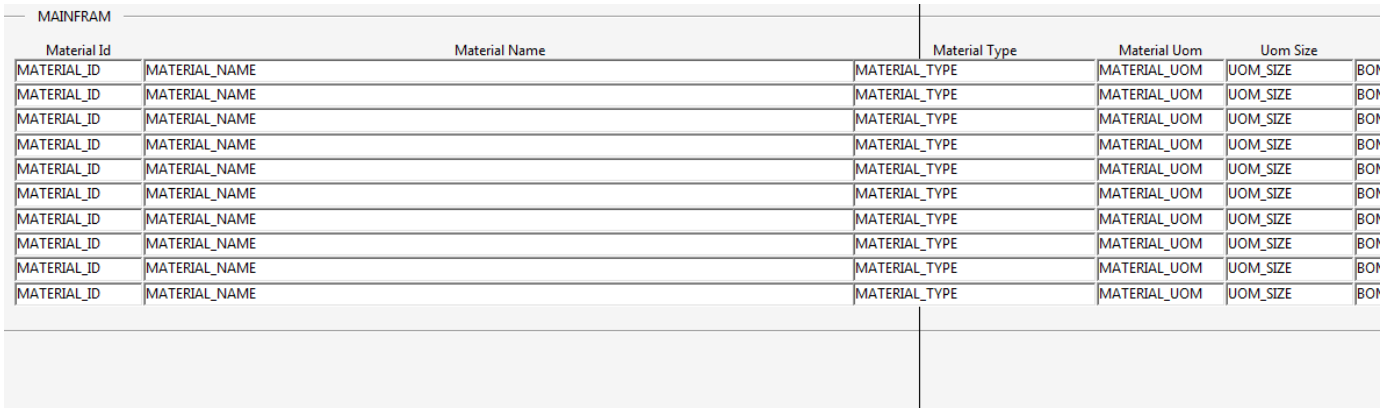


Click Finish



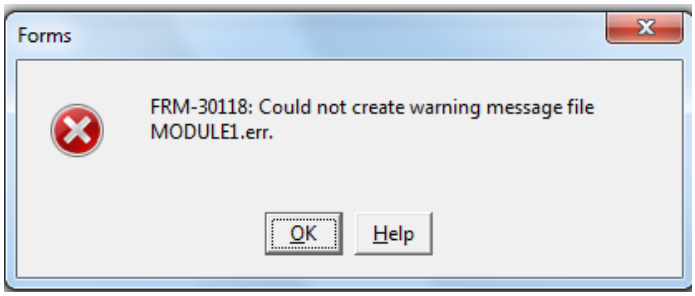
This will pop the layout editor. Expand the Layout Editor window.

Everything does not look so good here! The field Material Name is too long and has pushed other fields outside the boundary of the frame – MAINFRAME. We need to adjust this.

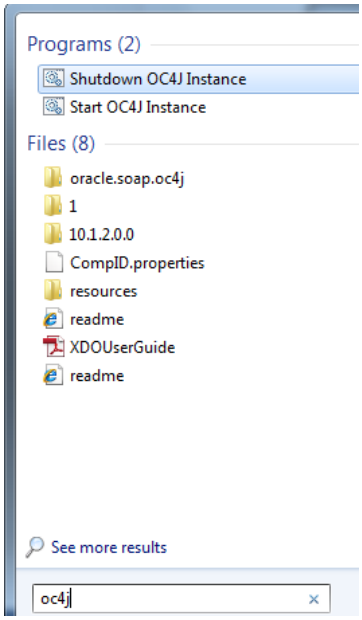


Well, let us try to run this and see how the running form looks like. Press the  button.

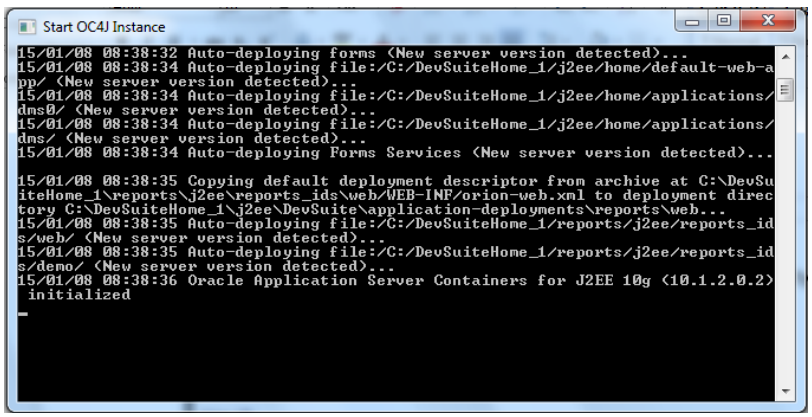
We will get the following error:



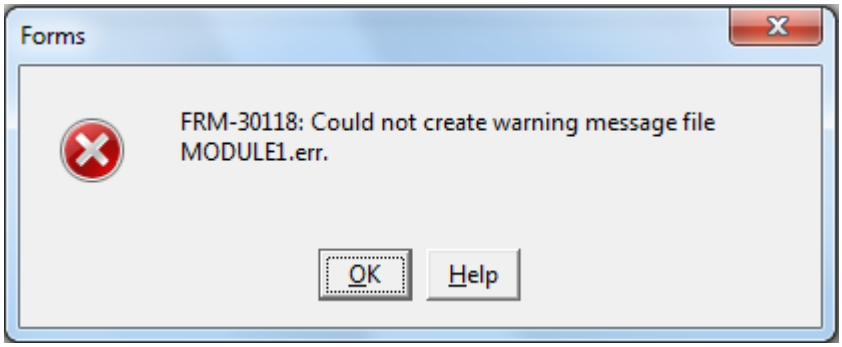
This is because the OC4J is not running. To run this Start > Run m type oc4j and select 'Start Oc4j Instance':



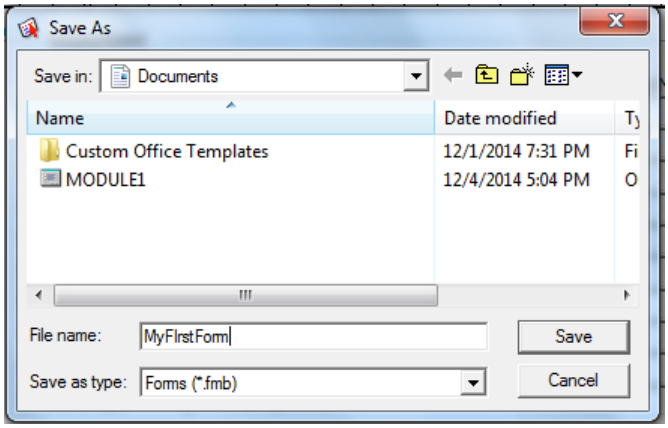
You will see a black windows like this: This is the Oc4J service, as long you have the forms running, this window cannot be closed.



Now try to run your form again, you will get another error:

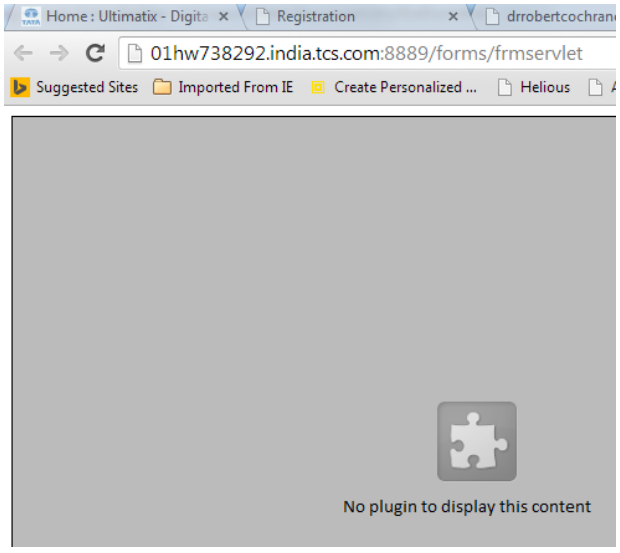


To resolve this, save the form you have crated, by File> Save As, and select the type as fmb



And run the form again.

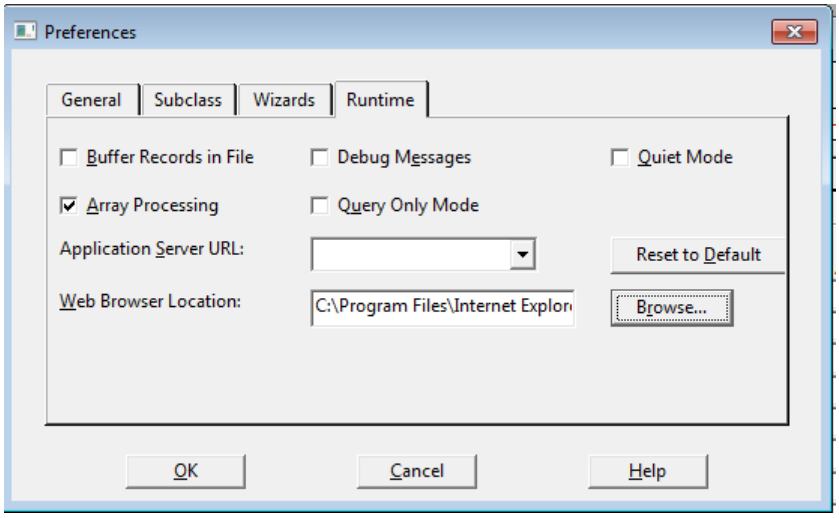
Well, we have another error:



Notice the following:

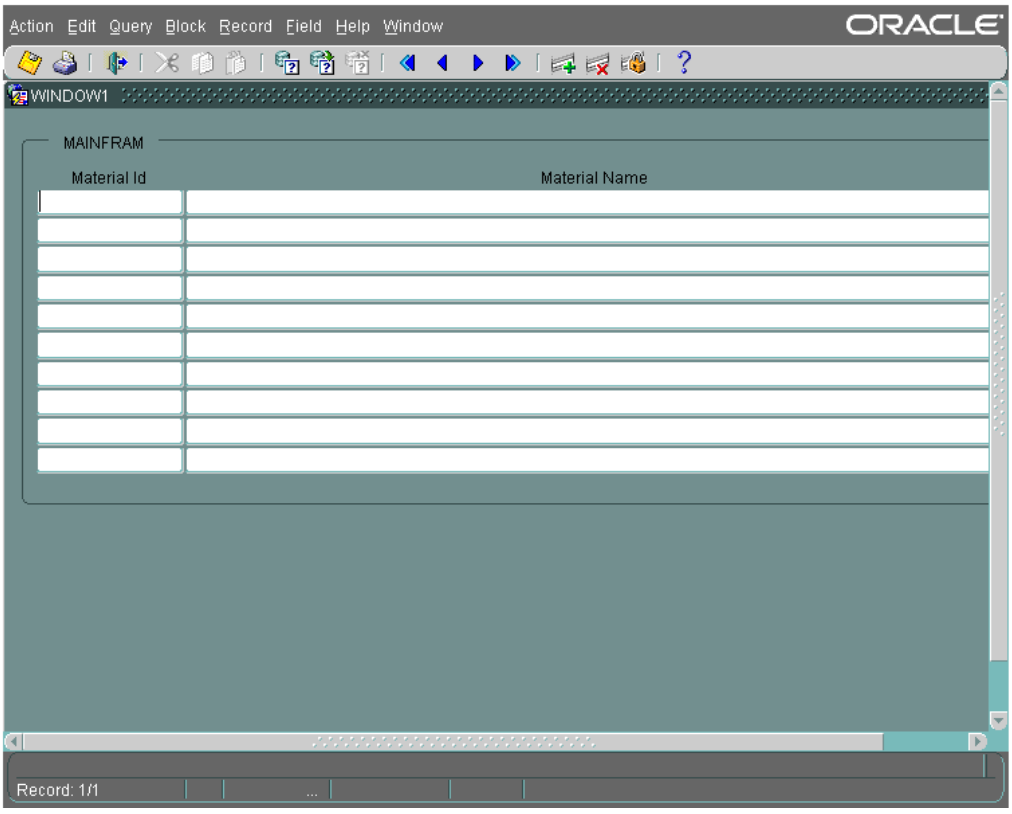
The forms builder runtime is trying to open the output in Chrome browser.
We need to tell the forms to use IE instead, we do it in this way:

1. Show the Web location browser- Edit> Preferences > Runtime

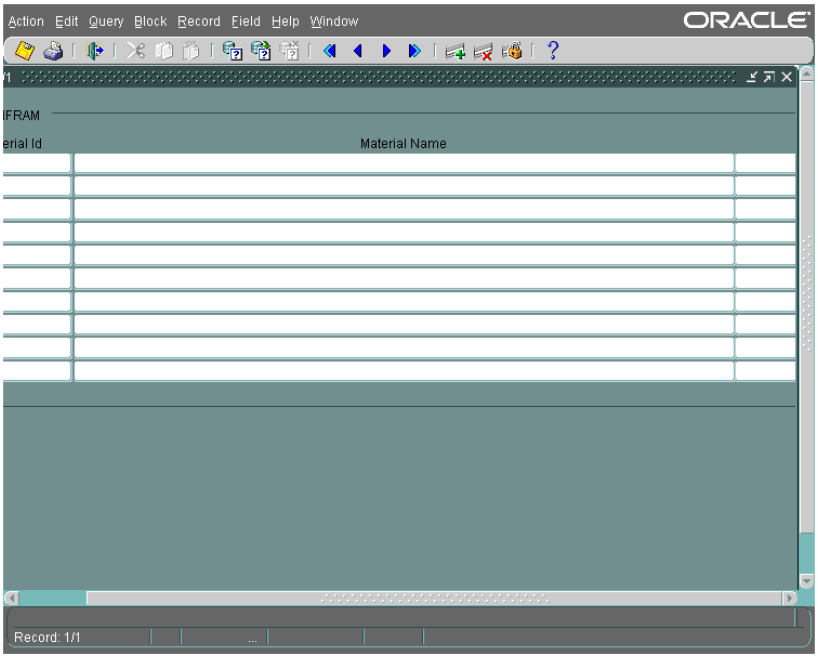


2. make sure Jinitiator is present in your machine (C:\Program Files\Oracle\JInitiator 1.3.1.30)

Run again, and you will see your form live:



Looks good, but there is a problem- You cannot see all the fields. You may think that it is hiding on the right, so use the scroll bar to see the right side, and you will see this:



Well, you still cannot see all the fields.

We will provide the solution to the same in the “Oracle Forms Study Guide - Day2.pdf”

Next Reading: “Oracle Forms Study Guide - Day2.pdf”