Oracle E-Biz is an ERP software. ERP means Enterprise Resource Planning, which means how to use and plan your company’s hardware resources in the best possible way to meet the Organization’s Software needs , while integrating with External system, that is systems outside the organization.

Around 80% of the ERP market is captured by SAP, which is E-Biz’s major competitor in the market, and around 20% is what Oracle E-Biz has captured in the global ERP market.

In any ERP implementation, 90% is B2B(Business to Business) programming, that is your organisation’s software communicating in both ways to softwares of another organization, which could be potentially a simple customer or a client or any other business partner.

Why Oracle E-Biz?

Oracle database is your backend, and a tool like SQL PLUS, or Oracle Forms is the Frontend.

Oracle 5.0 and 6.0 were purely UNIX-based flavours of the oracle database and could be shared by multiple users(multi-user systems) . To access these databases we used some tool like SQL PLUS or SQL FORMs or SQL Reports in the earlier days.

Oracle 7.0 was the first windows-based version of the Oracle database. It was only for windows platform. Later we had 7.1,7.2 and 7.3 versions which used D2K(Developer 2000) which included Oracle Forms, Oracle Reports.

Oracle was always a stand-alone database, that is Oracle Server and data resides on one single node. Hence, lets say we installed Oracle 7.3 on a Windows 95 OS machine(available in those days) and having Oracle D2K, it could not support client-server architecture.

Hence, Oracle 8.0 and 8i(OORDBMS which had http-services) were launched which could support client-server architecture. 8.0 however could only work on a company’s Intranet, not across the Internet and it used something called as SID for connectivity within the Intranet. Also, 8i had the http-service and the logical and physical database as separate entities. The HTTP-service allowed you to access the 8i database thru the web-browser using tool like iSQL Plus. Hence client server architecture was then supported on the internet. But, then in addition to the server having the database and D2K, even the clients connecting to the server had to have D2K(Example, from 1 PC which does not have MS WORD, you cannot connect to another machine which has MS WORD, and run MS WORD on the 1st machine.

Hence, some layer was required between the Database server and the client machine, to avoid having the D2K or any such software on the client machines. Hence, was launched the Oracle Application Server, which lies between(Middle tier) the client and the Oracle Database server.

In Microsoft products, there is no middle-ware/tier. Hence Microsoft has to depend upon other 3rd party tools like Tomcat, IIS to act as their middle-tier. But, Oracle has a backend, a frontend and a middleware(OAS).

Hence from Oracle 9i onwards, we also have something called Oracle 9i Application Server(Oracle 9i AS) as the middleware. Oracle 9i AS was a component of the E Biz version 11. E Biz 12 uses Oracle 10g AS as its middleware component

Thus, Oracle EBiz is actually using the Oracle Application Server capabilities to act as the middle-tier.

Single database system in E Biz

If an organization has various aspects of business like HR, Marketing, Sales, Inventory, etc, we needed to have a separate database for each one of them. Oracle says, have one single centralized database and have a middleware(E Biz) managed by the Oracle Apps DBA(sysdamin/sysadmin or Operations/welcome) and functional people(people who communicate with the end-users), and let the single centralized database be managed by the Oracle DBA, instead of having n-databases one each for each aspect of the organization. And let the various aspects of business like HR, Marketing, Sales, Inventory be implemented as modules of the E Biz. (schemas at the DB level)

Originally Oracle E Biz was called Oracle Financials as Oracle ERP or E- Biz was very strong in implementing the Finance module of an oragization.(Like wise SAP as an ERP is very string in implementing the Manufacturing module/aspect of an organisation).

E biz 12 focuses majorly on Java and it’s database(Oracle 10g database and partially Oracle 11g database are itself developed in Java), unlike earlier versions which were developed in C/C++ .

However, when we are going to use 1 single database for all the aspects of business, the hard-disk capacity to hold the database has to be very high. A machine may allow maximum say 4 harddisks on it. Hence was introduced the RAC(real application cluster) which allows to combine hard-disks of several machine on which the database can reside. RAC feature was launched with Oracle 9i. 10g implements the RAC feature along with the grid-structure, wherein we can have multiple instances of the database which are managed by the Enterprise Manager tool of Oracle 10g. This capability is also used in E Biz.

E Biz 12 uses an ETL tool to move data from legacy systems into the oracle database, with Oracle Warehouse Builder as an in-built feature/tool of Oracle 11g.

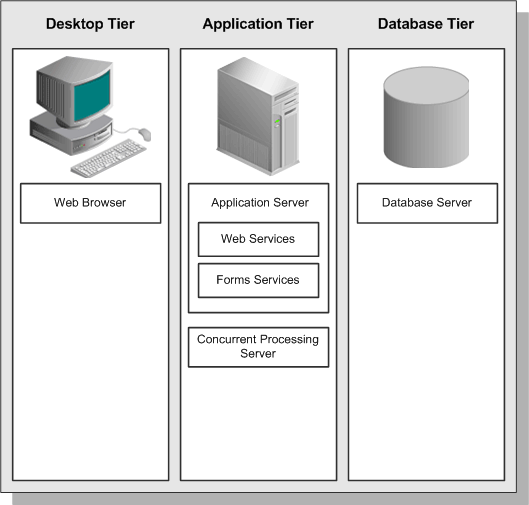
In E Biz 11, the database was 9i and the front end was 6i. But, with Forms/Report 6i, the forms/reports were stored locally on the D2K server

In E Biz 12, the Forms 10g/ADF frontend and/or the Reports 10g/XML publisher or Bi are used, and then the Forms and/or reports are stored in the AS. E Biz 12 works on both UNIX(Sun Solaris or RedHat Linux) and Windows.

In E Biz 12, at the database level, the individual aspects like HR, Marketing, Sales are implemented as individual schemas and end users are never allowed direct access to these schemas, as they are very sensitive and loss of any table/object as such can bring down the entire application. Hence, we have something called as the APPS schema, which contains view, synonyms for the table/objects of all other HR, Marketing, Sales schemas. It has all the necessary grants/privileges from the other HR, Marketing, Sales schemas.

In E Biz 12, at the individual module level, we have a GL(General Ledger) module, which interconnects all the other Finance, Marketing, Sales, Inventory modules for costing and finance purpose.

**Oracle E-Business Suite Architecture**



The three-tier architecture that comprises an Oracle E-Business Suite installation is made up of:

*database tier*, which supports and manages the Oracle database;

*application tier*, which supports and manages the various Oracle E-Business Suite components, and is sometimes known as the middle tier;

and the *desktop tier*, which provides the user interface(via an add-on component to a standard web browser). Ideally the node/machine from which you access the E-Business Suite.

The *Oracle E-Business Suite Architecture* is a framework for multi-tiered, distributed computing that supports Oracle E-Business Suite products(HR, SCM, etc). In this model, various *servers* or *services* are distributed among three *tiers*.

A server (or services) is a process or group of processes that runs on a single machine and provides a particular functionality. For example, *Web services or Web Server*  process HTTP requests, and *Forms services* *or Forms Server* process requests for activities related to Oracle Forms. The *Concurrent Processing server* supports data-intensive programs that run in the background.

A tier is a logical grouping of services, potentially spread across more than one physical machine. The three-tier architecture that comprises an Oracle E-Business Suite installation is made up of the *database tier*, which supports and manages the Oracle database; the *application tier*, which supports and manages the various Oracle E-Business Suite components, and is sometimes known as the middle tier; and the *desktop tier*, which provides the user interface via an add-on component to a standard web browser.

A machine is referred to as a *node*, particularly in the context of a group of computers that work closely together in a *cluster*. Each tier may consist of one or more nodes, and each node can potentially accommodate more than one tier. For example, the database can reside on the same node as one or more application tier components.

Centralizing the Oracle E-Business Suite software on the application tier eliminates the need to install and maintain application software on each desktop client PC which connect to the application tier, and also enables Oracle E-Business Suite to scale well with an increasing load of client PCs. The File system of the  *Application Tier, also know as Shared Application File System* (originally *Shared APPL\_TOP*) is such that we need to maintain only a single copy of the relevant Oracle E-Business Suite code, instead of a copy for every machine in the application tier, especially when the various servers of the middle tier are on different nodes or machines

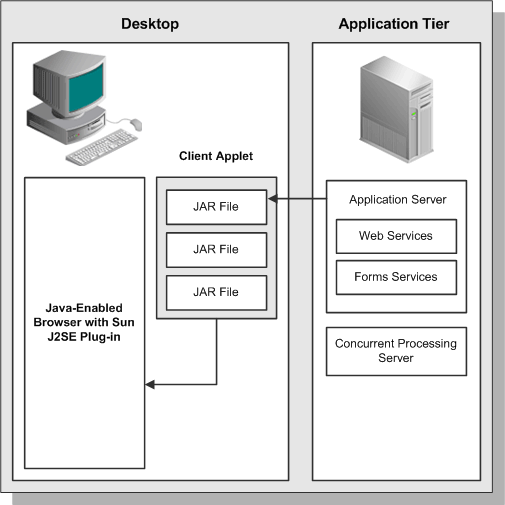
On the database tier, there is increasing use of *Oracle Real Application Clusters* (Oracle RAC), where multiple nodes support a single database instance to give greater availability and scalability.

The connection between the application tier and the desktop tier can operate successfully over a Wide Area Network (WAN). This is because the desktop and application tiers exchange a minimum amount of information, for example only field values that have changed. In a global operation with users at diverse locations, requiring less network traffic reduces telecommunications costs and improves response times.

**The Desktop Tier**

On the desktop tier, the client interface is provided through HTML for HTML-based applications, and via a Java applet in a Web browser for the traditional Forms-based applications.

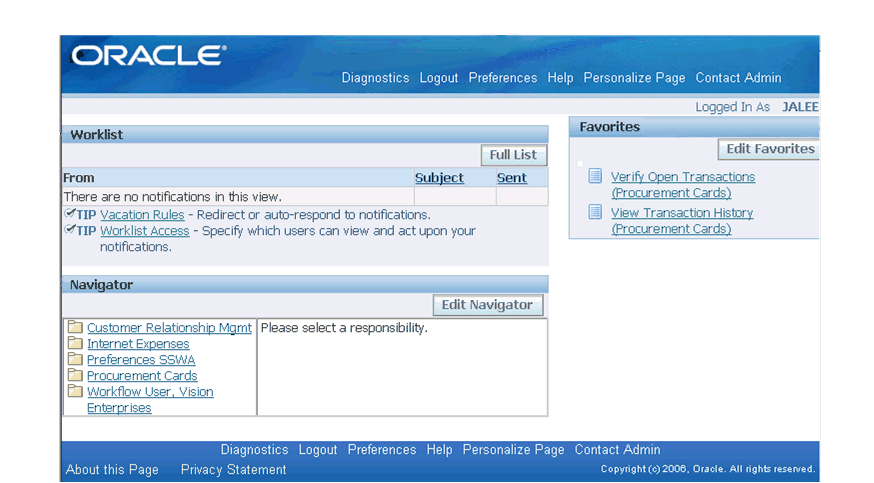
**Forms-based Desktop Tier Architecture**



You log in via the Oracle E-Business Suite Home Page on a desktop client web browser. The Home Page provides a single point of access to HTML-based applications, Forms-based applications, and Business Intelligence applications.

Once successfully logged in via the E-Business Suite Home Page, you are not prompted for your user name and password again, even if you navigate to other tools and products. Oracle E-Business Suite also retains preferences as you navigate through the system. For example, if you registered in the Home Page that German is your preferred language, this preference carries over whether you access Forms-based or HTML-based applications.

**Oracle E-Business Suite Home Page**



**Forms Client Applet**

The *Forms client applet* is a general-purpose presentation applet that supports all Oracle E-Business Suite Forms-based products. The Forms client applet is packaged as a collection of *Java Archive* (JAR) files. The JAR files contain all Java classes required to run the presentation layer of Oracle E-Business Suite forms.

**Desktop Java Client**

The Forms client applet must run within a *Java Virtual Machine (JVM)* on the desktop node. The *Sun JRE Plug-in* component allows use of the Oracle JVM on web clients, instead of the browser's own JVM. This component is implemented as a standard browser plug-in.

In the traditional, Forms-based Oracle E-Business Suite environment, the JVM was run as part of the standard Oracle E-Business Suite sign-on process. Now, the JVM (JRE Plug-in in Release 12) is only invoked when a user chooses to access functions that require it, such as running a form. If the Sun JRE Plug-in has not been installed, the browser prompts the user to download the required installation executable. After you download and install the plug-in, you will be able to run Forms-based applications.

The Forms client applet and commonly used JAR files are downloaded from the Web server at the beginning of the client's first session. Less commonly used JAR files are downloaded as needed. All downloaded JAR files are cached locally on the client, ready for future sessions. This eliminates the network traffic that would be involved in downloading them whenever they were required. All updates to JAR files are installed on the application tier and downloaded to the client automatically, via the caching mechanism

## The Application Tier

The *application tier* has a dual role:

hosting the various servers and service groups that process the business logic

and managing communication between the desktop tier and the database tier.

This tier is sometimes referred to as the *middle tier*.

Three servers or service groups comprise the basic application tier for Oracle E-Business Suite:

* Web services
* Forms services
* Concurrent Processing server

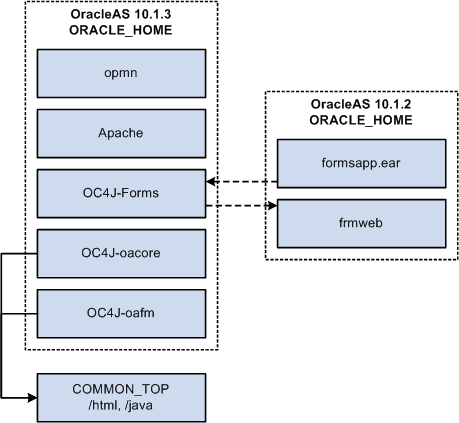
In Release 12, Web and Forms services are provided by *Oracle Application Server*(OracleAS) 10g.

It is advisable to avoid using a mixture of different platforms(Operating Systems for the different nodes of your middle or application tier) on your application tier. This makes maintenance easier, since only one set of patches needs to be downloaded. There is no concept of a separate Administration server in Release 12. By default, patching/upgradation can be undertaken from any application tier node.

### Use of Two Oracle Application Server ORACLE\_HOMEs in Release 12

Two different Oracle Application Server (OracleAS 10g) HOMES are used in Oracle E-Business Suite Release 12. This enables Oracle E-Business Suite to take advantage of the latest Oracle technologies.

**Relationship between the two Application Server ORACLE\_HOMEs**



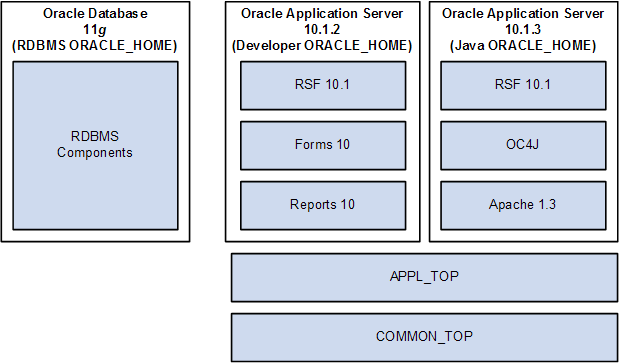
Notable features of this architecture include:

* The latest version of Oracle Containers for Java (OC4J), the successor to JServ, is included in Oracle Application Server 10.1.3.
* All major services are started on the OracleAS 10.1.3 ORACLE\_HOME.
* The Oracle E-Business Suite modules (packaged in the file formsapp.ear) are deployed into the OC4J-Forms instance running on the OracleAS 10.1.3 ORACLE\_HOME, while the frmweb executable is invoked on the OracleAS 10.1.2 ORACLE\_HOME.
* oacore OC4J - Supports framework based applications.

forms OC4J - Supports forms based applications

oafm OC4J - expands to Oracle Application Fusion Middleware - for mapviewer, webservices, ascontrol

**Database and Oracle Application Server ORACLE\_HOMEs**



Notable features of this high-level architecture include:

* The Oracle Application Server 10.1.2 ORACLE\_HOME (sometimes referred to as the Tools, C, or Developer ORACLE\_HOME) replaces the 8.0.6 ORACLE\_HOME provided by Oracle9*i* Application Server 1.0.2.2.2 in Release 11*i*.
* The Oracle Application Server 10.1.3 ORACLE\_HOME (sometimes referred to as the Web or Java ORACLE\_HOME) replaces the 8.1.7-based ORACLE\_HOME provided by Oracle9*i* Application Server 1.0.2.2.2 in Release 11*i*.

## Web Services(OracleAS 10.1.3)

The Web services component of Oracle Application Server processes requests received over the network from the desktop clients, and includes the following components:

* Web Listener (Oracle HTTP Server powered by Apache)
* Java Servlet Engine [OC4J(OC4J-Forms or OC4J-oafm or OC4J-oacore)]
* Oracle Process Manager (OPMN)

The Web listener component of the Oracle HTTP server accepts incoming HTTP requests (for particular URLs) from client browsers, and routes the requests to the appropriate OC4J container(OC4J-Forms or OC4J-oafm or OC4J-oacore)

If possible, the Web server services the requests itself, for example by returning the HTML to construct a simple Web page. If the page referenced by the URL needs advanced processing, the listener passes the request on to the *servlet engine*, which contacts the database server as needed.

### HTML-Based Applications and the Oracle Application Framework using OC4J-oacore

The Oracle HTML-based applications (originally known as Self-Service applications) have the following characteristics:

* Do not use Oracle Forms for the interface
* Are designed in pure HTML and JavaScript
* Dynamically generate HTML pages by executing Java code
* Use a metadata dictionary for flexible layout
* Operate by direct connection to the Web server

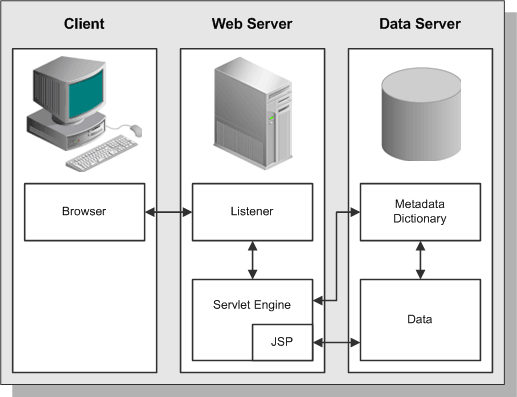
The *Oracle Application Framework* is the development platform for HTML-based applications. It consists of a Java-based application tier framework and associated services, designed to facilitate the rapid deployment of HTML-based applications.

Notable Oracle Application Framework components include:

* *Business Components for Java* (BC4J), included in Oracle JDeveloper, is used to create Java business components for representing business logic. It also provides a mechanism for mapping relational tables to Java objects, and allows the separation of the application business logic from the user interface.
* *AOL/J* supplies the Oracle Application Framework with underlying security and applications Java services. It provides the Oracle Application Framework with its connection to the database, and with application-specific functionality such as flexfields.

The Framework-based applications logic is controlled by procedures that execute through the Java servlet engine, which is provided by the Apache JServ module. The servlet engine uses the metadata dictionary in constructing the Framework UI.

**Figure shows HTML-Based Applications Architecture**

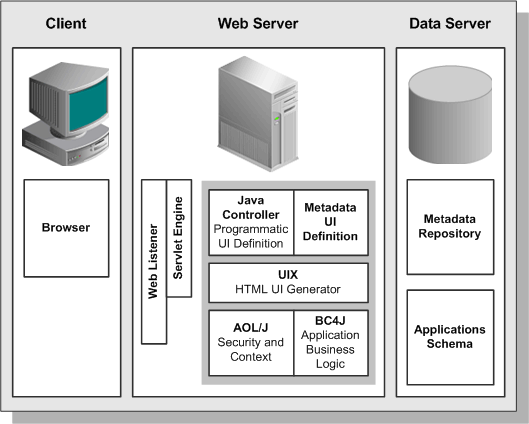


**Java Servlet Access with HTML-Based Applications**

An HTML-based Applications module uses the following access path:

1. The user clicks the hyperlink of a function from a browser.
2. The browser makes a URL request to the Web listener.
3. The Web listener contacts the Servlet engine (OC4J-oacore), where it runs a JSP.
4. The JSP obtains the content from the Oracle E-Business Suite tables and uses information from the metadata dictionary to construct the HTML page.
5. The resulting HTML page is passed back to the browser, via the Web server.

**Oracle Application Framework Architecture**



### Oracle Application Framework Processing Details

The following is a more detailed explanation of how the JSP obtains the content from the Oracle E-Business Suite tables and uses information from the metadata dictionary to construct the HTML page.

1. AOL/J validates user access to the page.
2. The page definition (metadata UI definition) is loaded from the metadata repository on the database tier into the application tier.
3. The BC4J objects that contain the application logic and access to the database are instantiated.
4. The Java Controller programmatically manipulates the page definition as necessary, based on dynamic UI rules.
5. UIX (HTML UI Generator) interprets the page definition, creates the corresponding HTML in accordance with UI standards, and sends the page to the browser.

## Forms Services

By default, Forms services in Oracle E-Business Suite Release 12 are provided by the *Forms listener servlet(OC4J-Forms)*, benefits of using the Forms listener servlet include:

* Ability to re-establish dropped network connections
* Easier firewall/proxy server configuration
* More robust and secure deployment over the Internet

### Forms Listener Servlet Architecture

The Forms listener servlet*(OC4J-Forms)*, is a Java servlet that delivers the ability to run Oracle Forms applications over HTTP or HTTPS connections. It hosts the Oracle E-Business Suite forms and associated runtime engine, mediating the communication between the desktop client and the Oracle database server, displaying client screens, and initiating changes in the database according to user actions.

The Forms listener servlet caches data and provides it to the client as needed, for example when scrolling through multiple order lines that exceed the limitations of a single screen.

The Forms listener servlet can communicate with the desktop client using either a standard HTTP network connection or secure HTTPS network connection. The Forms listener servlet communicates with the Oracle database server using the *Oracle Net* networking infrastructure. In contrast, Forms services (formerly known as Forms server) on the 10.1.2 home(Developer Home of the Oracle Application Server)communicates with the desktop client using the TCP/IP network protocol, on top of which it layers its own protocol.

The Forms listener servlet manages the creation of a Forms runtime process for each client, as well as network communications between the client and its associated Forms runtime process. The client sends HTTP requests and receives HTTP responses from the Web services, which acts as the network endpoint for the client.

Although the Forms listener servlet (OC4J-Forms instance) runs on the OracleAS 10.1.3 ORACLE\_HOME, the frmweb executable runs on the OracleAS 10.1.2 ORACLE\_HOME.

**Concurrent Processing Server**

User interactions with Oracle E-Business Suite data can be conducted via HTML-based applications or the more traditional Forms-based applications. However, there are also reporting programs and data updating programs that need to run either periodically, or on an ad hoc basis. These programs, which run in the background while users continue to work on other tasks, may require a large number of data-intensive computations, and are run using the *Concurrent Processing* architecture. Concurrent Processing is an Oracle E-Business Suite feature that allows these non–interactive and potentially long-running functions to be executed efficiently alongside interactive operations. It uses operating system facilities to enable background scheduling of data- or resource-intensive jobs, via a set of programs and forms. To ensure that resource-intensive concurrent processing operations do not interfere with interactive operations, they are run on a specialized server, the *Concurrent Processing server*.

Processes that run on the Concurrent Processing server are called *concurrent requests*. When you submit such a request, either through HTML-based or Forms-based applications, a row is inserted into a database table specifying the program to be run. A *concurrent manager* then reads the applicable requests in the table, and starts the associated concurrent program.

**Concurrent Manager Characteristics**

Concurrent managers are fundamental to concurrent processing. Acting as a job scheduling and execution system, a concurrent manager:

* Is an executable that is registered as a program library within Oracle E-Business Suite, and which runs in its own operating system process
* Runs operating system processes called *target processes* (often referred to as *workers*), each of which can start one concurrent program at a time
* Can optionally run an *immediate program* that runs as part of the concurrent manager’s own operating system process
* Can be allowed to run any concurrent program, or be specialized to run certain programs
* Operates during the days and times defined by a *work shift*

**Concurrent Processing Architecture**

In Concurrent Processing, programs are run as operating system background processes. These programs may be written using a variety of Oracle tools, programming languages-executables, or the host operating system scripting language.

A concurrent program that runs in the concurrent manager's own operating system process is known as an *immediate program*. Immediate programs run as a function within the concurrent manager’s program library. Examples include PL/SQL programs. In contrast, a concurrent program that runs in a child process of the concurrent manager process is known as a *spawned program*. Examples include SQL programs, SQL Loader programs, Oracle Reports programs, spawned C programs, and host language programs such as UNIX shell scripts or Windows command files.

All reports are run through the Concurrent Processing server manager via the rwrun executable, which spawns an in-process server. This is in contrast to earlier releases of Oracle E-Business Suite, which used the now-obsolete Reports server.

While C programs can be run as immediate programs, it is advisable to run them as spawned programs. This simplifies maintenance, without introducing any disadvantages.

A concurrent request has a life cycle, which consists of three or possibly four phases:

|  |  |
| --- | --- |
| **Table 1-1 Concurrent Request Life Cycle** | |
| **Phase** | **Activity** |
| Pending | The request is waiting to be run |
| Running | The request is running |
| Completed | The request has finished |
| Inactive | The request cannot be run |

A *concurrent program library* contains concurrent programs that can be called by a concurrent manager. An important example is the Oracle Application Object Library program library (FNDLIBR), which contains Oracle E-Business Suite immediate concurrent programs, and is assigned to the standard concurrent manager. Although each concurrent manager can only run immediate concurrent programs from its own concurrent program library, it can also run spawned or Oracle tool concurrent programs.

Various database tables are employed by the concurrent processing architecture:

|  |  |
| --- | --- |
| **Table 1-2 Concurrent Processing Database Tables** | |
| **Table** | **Content** |
| FND\_CONCURRENT\_REQUESTS | Details of user requests, including status, start date, and completion date |
| FND\_CONCURRENT\_PROGRAMS | Details of concurrent programs, including execution method, whether the program is constrained, and whether it must be run alone. |
| FND\_CONCURRENT\_PROCESSES | Cross-references between concurrent requests and queues, and a history of concurrent manager processes |
| FND\_CONCURRENT\_QUEUES | Information about each of the concurrent manager queues |

**Caution:** Do not update these tables manually. You can (subject to your organization's archiving requirements) periodically run the "Purge Concurrent Requests and/or manager data" program to prevent these tables growing too large.

## Application Tier Administration

In Oracle E-Business Suite Release 12, any application tier node can be used to carry out the following operations:

* **Applying database patches to Oracle E-Business Suite**

In general, *Oracle E-Business Suite patches* consist of files and scripts that update the file system and database objects. You use the *AutoPatch* utility (*adpatch*) to perform these updates.

* **Maintaining Oracle E-Business Suite data**

Some features require updates to the tables and schemas they use. The *AD Administration* utility (*adadmin*) enables you to carry out this and various other file system and database maintenance tasks.

## The Database Tier

The database tier contains the Oracle database server that stores and manages all the data maintained by Oracle E-Business Suite. This includes the various types of file in which the tables, indexes, and other database objects for your system physically reside, as well as the database executables. The database also stores the Oracle E-Business Suite online help information.

The database server communicates with the services and servers on the application tier, which mediate the communications between the database and the clients: there is no direct communication between the database and clients.

### Using a Mixed Platform Architecture

The Oracle database server is sometimes available on platforms where Oracle E-Business Suite is not currently certified. In such a case, it may be possible to utilize a *mixed platform architecture*, where the database is installed on one platform and the application tier on another. (In Release 11*i*, this was referred to as a *split configuration*.)

This type of deployment can enable the database to utilize the specific features offered by a particular platform (such as a 64-bit architecture). It can also allow the application tier to be managed in a more cost-effective way.

**The Oracle E-Business Suite Technology Layer**

The Oracle E-Business Suite technology layer lies between the Oracle E-Business Suite technology stack and the Oracle E-Business Suite product-specific modules. It provides features common to all Oracle E-Business Suite products.

Products in the Oracle E-Business Suite technology layer include:

* Oracle Applications DBA (AD)
* Oracle Application Object Library (FND)
* Oracle Applications Utilities (AU)
* Oracle Common Modules (AK)
* Oracle Workflow (WF)
* Oracle Alert (ALR)
* Oracle Application Framework (FWK)
* Oracle XML Publisher (XDO)

## Oracle Applications DBA (AD)

The Oracle Applications DBA product provides a set of tools for administration of the Oracle E-Business Suite file system and database. AD tools are used for installing, upgrading, maintaining, and patching the Oracle E-Business Suite system.

## Oracle Common Modules (AK)

AK is an active data dictionary that enables you to define Oracle E-Business Suite components for the HTML-based applications. The *Oracle Common Modules* can be used to develop inquiry applications for the HTML-based applications, without the need for any programming.

## Oracle Applications Utilities (AU)

The Applications Utilities (AU) component is used to maintain the Oracle E-Business Suite system. AU hosts a collection of files copied from other products. This allows generating on-site classes of files such as Forms and reports. Generating forms or reports may require access to shared PL/SQL libraries, so these files are copied to AU\_TOP as well.

## Oracle Application Object Library (FND)

The Oracle Application Object Library is a key component of the Oracle E-Business Suite technology layer. It consists of a collection of reusable code, programs, and database objects that provides common functionality across all products. Oracle Application Object Library offers many features to make system administration easier, such as security setup and maintenance, and management of concurrent processing.

### End User Features

Oracle Application Object Library includes several features that help provide uniformity of function across the various Oracle E-Business Suite products.

**Standard User Interface**

Oracle Application Object Library supports the integration of Oracle E-Business Suite by providing standardized functionality and capabilities across all products so that the look and feel remains the same from product to product.

**Shared Flexfield value sets**

Flexfields allow the entry of certain important information to be standardized across all products. One example is the Accounting Flexfield, which is used by Financials products and Manufacturing products.

**Standard Report Submission (SRS)**

The procedure to submit a background report to the concurrent manager using SRS is the same, regardless of the product that owns the report. SRS takes advantage of shared flexfield value sets.

**Online Help**

The presentation of Online Help is also standardized across all Oracle E-Business Suite products

### Developer Features

Oracle Application Object Library provides many features for developers creating custom forms, reports, or programs that interface with Oracle E-Business Suite:

### Features for System Administrators

Oracle Application Object Library provides many features to simplify administration of Oracle E-Business Suite, enabling the system administrator to carry out routine tasks quickly and easily. These features include:

* Registering new Oracle E-Business Suite users, and giving them access to only those Forms, functions, and reports they need to do their jobs.
* Deciding which users have access to each product, and within a product, which Forms, functions, and reports a user can access.
* Monitoring what users do, and when, via comprehensive auditing capabilities.
* Setting user and system *profiles* to modify the look and behavior of Oracle E-Business Suite products; profiles can be set at site, application, responsibility, and user levels.
* Monitoring and controlling concurrent processing using interfaces such as Oracle Applications Manager (OAM).

### Oracle Application Object Library Security

Oracle Application Object Library controls access to the data in Oracle E-Business Suite via user sign-ons and responsibilities. Each user must have a valid user name and password to gain access to Oracle E-Business Suite.

A *responsibility* is a level of authority in Oracle E-Business Suite that lets users access only those functions and data appropriate to their roles in the organization. For example, responsibilities may be used to allow access to a specific product, ledger, operating unit, or to a restricted list of windows, functions, reports, and groups of products, or data groups.

Note that the Forms available from the navigation menus vary by responsibility. For example, the Purchasing User navigation menu does not include all the forms that are available to the Purchasing Superuser navigation menu.

When you install Oracle E-Business Suite, a standard user called SYSADMIN is created for you. Several default responsibilities are also created. Since the SYSADMIN sign-on is automatically assigned to the System Administrator responsibility, you can use SYSADMIN to create new user signons and assign them to responsibilities. You can also create any custom responsibilities you need.

## Oracle Workflow (WF)

Oracle Workflow delivers a complete workflow management system that supports business process based integration. Oracle Workflow also provides an infrastructure for the enterprise-wide communication of data related to defined business events, providing the capabilities needed to:

* Manage enterprise business processes that may span trading partners
* Support standard and personalized business rules
* Streamline and automate transaction flows
* Manage exceptions without manual intervention

### Workflow Components

*Oracle Workflow Builder* provides a graphical drag and drop process designer. You can create and evolve business processes to incorporate existing business practices between your organization and customers or suppliers, without modifying existing business processes and without changing applications code.

The *Workflow Engine*, embedded in the Oracle database, implements process definitions at runtime.

## Oracle Alert (ALR)

Oracle Alert (ALR) allows you to email system notifications to users when an exception or event has occurred. Some products are delivered with predefined alerts, which can be used to notify users about specified database exceptions as they occur, and perform routine tasks automatically according to a schedule you define.

For example, you can configure Oracle Alert to send an email to key database administrators when a tablespace in the Oracle E-Business Suite database does not have adequate free space.

## Oracle XML Publisher (XDO)

Oracle XML Publisher is a Java-based product based on the World Wide Web Consortium (W3C) *Extensible Stylesheet Language* (XSL). XML Publisher uses *data definitions* and *templates* to produce output reports in the desired format. A data definition is a data source (or a combination of data sources) that either is XML or can produce XML. Examples include output from concurrent programs and Web services. A template is a report definition, which sets out how a report should look. The template layout can be user-specified. Supported templates currently include RTF, PDF Forms, and XSL.

## Oracle Configuration Manager

*Oracle Configuration Manager* (OCM) is a component that is designed to facilitate support for your Oracle products. Use of Oracle Configuration Manager is optional, but recommended.

A lightweight agent that consumes minimal CPU resources, OCM supports automatic discovery of installed components and configuration information, and provides continuous tracking of key Oracle and system statistics of the machine it is running on.

## Oracle E-Business Suite Patch Nomenclature

Oracle E-Business Suite Release 12 includes numerous products that are designed to serve a wide variety of needs in a consistent and complementary way. All have abbreviated names or two or more usually three letters. These short names are used in special patch names (see below) and other places. In general, the short names relate to the full product name. Common examples include Applications Technology (ATG), Business Intelligence (BIS), Financials (FIN), Human Resources (HR), and Supply Chain Management (SCM).