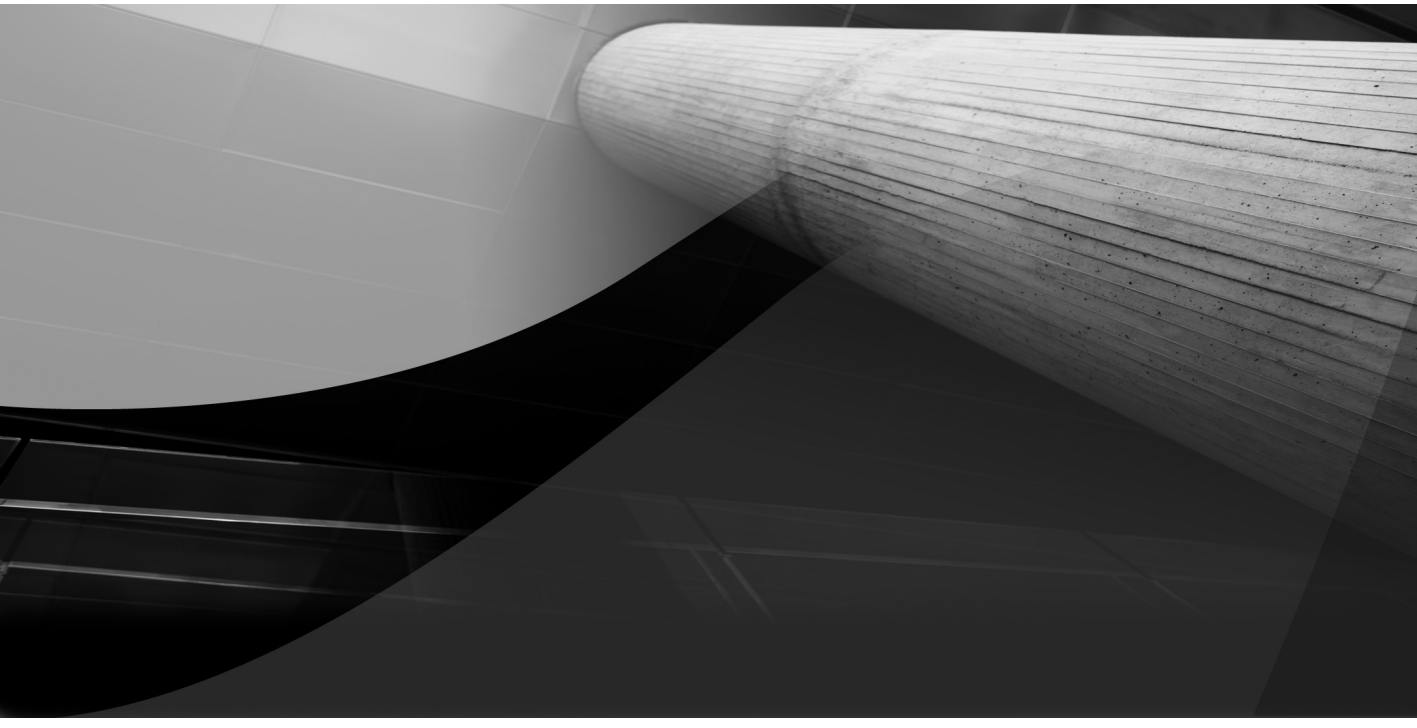


CHAPTER 1

Introduction to Oracle E-Business Suite

n this introductory chapter, we'll give a high level functional overview of Oracle E-Business Suite from an application developer's point of view. This chapter is primarily aimed at the readers who are familiar with Oracle tools but new to E-Business Suite; those who have already gained experience in working with E-Business Suite can safely skip this chapter.

We also look at what options are available to implementation teams and developers to change the standard product features, and later in the chapter we briefly discuss the concept of E-Business Suite environments.

At the end of this chapter we look at how information is shared and reused within different modules in Oracle Applications in order to highlight the importance of data sharing between different modules within E-Business Suite.

What Is Oracle E-Business Suite?

Oracle E-Business Suite is a software package that allows organizations to manage key business processes; it is known on the market by various names such as Oracle Enterprise Resource Planning (ERP), Oracle Apps, Oracle Applications, Oracle Financials, e-Biz and EBS (E-Business Suite). In this book we refer to it as either E-Business Suite, or Oracle Applications.

In the past, it was a common practice for businesses and organizations to develop in-house software to automate their business processes. Most of the software that was developed in-house largely matched the precise needs of the business. However, the fundamental business flows and processes such as accounting, procurement, human resource/employee management, and order management are based on common principles across all organizations. For example, most organizations require a system to make purchases from suppliers and a system to make payments to the suppliers, events known as transactions that need to be accounted for in the financial reporting. Enterprise Resource Planning (ERP) software prepackages different types of these functionalities into out-of-the-box software package, so that customers who purchase such software packages do not have to develop the same software applications time and again.

Product Families

Oracle E-Business Suite is a product offering that covers almost all of the business flows widely used in most organizations. Businesses can implement as many modules as necessary due to the modular but still integrated nature of the E-Business Suite architecture. This allows unified information to be available across the enterprise; it also reduces information technology (IT) expenses and helps run business more efficiently.

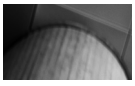
On the contrary, managing heterogeneous software solutions developed in-house that use different systems and technologies can be extremely costly and complex. Any time you update one system, you must go back and review all the integration points and potentially update the interfaces between the systems. Oracle E-Business Suite is engineered to work as an integrated system on a common IT infrastructure. You can directly pass information from one application to another without incurring incremental integration costs.

The product offering in E-Business Suite is organized into product families. Some of the key product families are as follows:

- Financials
- Procurement
- Customer Relationship Management (CRM)
- Project Management
- Supply Chain Planning and Management
- Discrete Manufacturing
- Process Manufacturing
- Order Management
- Human Resources Management System (HRMS)
- Applications Technology

In E-Business Suite, each product family usually consists of individual applications. For example, some of the applications that make up the Oracle Financials product family are General Ledger, Payables, Receivables, Cash Management, iReceivables, iExpenses, and others. It is beyond the scope of this book to cover the functionality of products such as General Ledger, Oracle Purchasing, and the like. There is a wealth of information about the functionality of E-Business Suite products publicly available, and we suggest the following resources for further reading:

- **Oracle Technology Network (Documentation section)** www.oracle.com/technology/documentation/applications.html
- **Oracle E-Business Suite** www.oracle.com/applications/e-business-suite.html
- **Oracle Metalink (requires registration)** metalink.oracle.com



NOTE
Throughout this book, we'll sometimes refer to E-Business Suite applications as modules. The terms application *and* module *will be used interchangeably.*

Professional User Interface

When the Oracle ERP product was initially launched, the screens were built in character mode. The end users interacted with the system through dumb terminals, which provided a character-based interface that connected to the back end server. Both Oracle Forms (then known as SQL*Forms) and Oracle Database were run at the back end tier. Initially, the R10.7 version of Oracle Applications ran in character mode, as did all the previous releases. However, when Oracle released its GUI version called SmartClient, the SmartClient screens were built with Oracle Forms 4.5 and ran at the desktop client tier, accessing the database over the network. Although SmartClient provided a better user experience, it was difficult to maintain, as software updates needed to be distributed on every individual client desktop. Last in that release, Oracle announced R10.7 NCA (Network Computing Architecture), which was an attempt to integrate the latest web technologies into Oracle's business applications using three-tier architecture, including database and application servers; end users interacted with the system using the browser from their client desktops. The latest releases of E-Business Suite, R11i and R12, are also based on multi-tier architecture, and the details will be covered in the next chapter.

Nowadays, in the latest releases of E-Business Suite R11i and R12, we refer to the professional user interface as an interface that is built with the Oracle Forms developer tool. Such Forms-based screens run within a Java applet at the client desktops, and in their appearance and behavior they are similar to desktop applications. Office personnel who often performs data entry tasks usually prefer using this type of user interface as it allows speedy data entry.

Web User Interface

As mentioned in the previous section, most of the screens in Oracle E-Business Suite were initially developed using Oracle Forms. However, over the last few years, Oracle has started to deliver new screens using pure web-based technology. These web-based screens do not run within a Java applet, unlike Forms-based screens. Instead, the HTML-based screens are run with a browser such as Firefox or Internet Explorer. Oracle initially started developing HTML-based pages in E-Business Suite primarily to provide a light footprint application or Self-Service–based applications. Here are some examples of the Self-Service applications:

- **HR Self-Service** End users maintain their own personal records, such as name, address, and so on.
- **iProcurement** Users create requisitions to buy goods such as stationery by themselves directly rather then having a central purchasing team creating that order for them.
- **iRecruitment** Users apply for a different job internally within their organization.
- **iExpenses** Employees submit their expenses for approval via a web interface.

The reasons that justify the broad adoption of an HTML-based interface is ever increasing; here we list just a few of them:

- Commitment to the open industry standards usually leads to the increased product interoperability.
- A pure HTML-based web application is lightweight and it runs without the need for a Java applet in the browser.
- An adoption of the new components and emerging technologies such as AJAX, Rich Internet Applications (RIA), and others ensures a better end user experience.

As a result of the preceding factors, even the new back office screens are now being developed as HTML-based pages using Oracle Application Framework (OA Framework). The sophisticated user interface features that were previously offered only through Oracle Forms are increasingly becoming available to HTML-based screens that run exclusively within desktop browsers, without the need for Java applets.

Nowadays, Oracle E-Business Suite developers find themselves working with both Oracle Forms and OA Framework, as the current releases (Release 11i and Release 12) contain a mixture of screens using both the technologies. Later in the book, we cover both Oracle Forms in Chapter 6 and E-Business Suite Oracle Applications Framework development techniques in Chapter 9.

Configurations, Personalizations, Extensions, and Customizations

Oracle E-Business Suite was designed and developed to take into consideration various standard business flows that are common to most organizations. However, each business can have its own unique requirements. For example, a company may want to allow all of its employees to make purchases up to $10 without

having such purchases approved. Another company may have a business rule that each employee's approval limit depends on his or her position within the organization. Oracle E-Business Suite is a package that has to meet not only the needs of both these types of companies, but also the needs of numerous other companies that may have a completely different set of requirements and business needs. That's why Oracle E-Business Suite has been developed in a configurable manner, so that each customer can buy this package and configure it to meet his or her specific business requirements. However, if business requirements cannot be met purely by using setup and configuration options, implementations have to resort to other options such as system personalizations, extensions, and customizations, which may or may not require custom code to be written by an E-Business Suite technical developer.

Configurations

E-Business Suite is an off the shelf software package that is both configurable and extensible. Changes are mostly made to ERP products by means of setup and configurations. Performing a setup usually means making changes to the product, without writing computer programs. System or product configuration is normally performed by functional analysts.

Personalizations

In E-Business Suite, the underlying technologies that render the user interface at presentation layer allow system implementers and end users to declaratively personalize the content of application forms and web pages. If business needs cannot be met by system configuration and setup, this is the first option to look at as it provides the safest way to change the system.

The major technologies that enable user personalizations in E-Business Suite are Oracle Forms and Oracle Application Framework (OAF), often referred to as Forms Personalizations and OA Framework Personalizations. We cover both Forms and OAF personalizations later in this book in chapters that cover the corresponding tools (Oracle Forms in Chapter 6 and OAF Personalizations in Chapter 9).

Customizations and Extensions

If, due to the generic nature of the product or any other reason, certain business requirements cannot be met through the product configuration and personalization, the technical development team is required either to extend the existing product functionality or introduce completely new custom modules that seamlessly integrate with the standard product and functionality. Depending on the underlying technology, both customization and extension terms are often used interchangeably, and usually they mean one thing: extending the product functionality by means of writing custom code that is either tightly or loosely coupled with E-Business Suite applications code and, in some cases, even completely decoupled from product code.

E-Business Suite developers are advised to err on the side of caution when dealing with customizations and extensions. It is important to stress that Oracle Support Services (OSS) do not support custom code logic in customizations developed to extend the standard product functionality. The general rule of thumb is that if something is not documented, then it is not supported by OSS. Most of the tools used by developers to build product customizations have corresponding support guidelines published on Metalink. Here are some examples:

Note 578466.1 Oracle Workflow Customization Policy Clarification

Note 395441.1 Oracle Application Framework Support Guidelines for Customers Release 12

More generic policy regarding the customizations is explained in Metalink Note 209603.1: Oracle Support Services Policy Regarding Customizations. Ultimately, if unsure about any aspect of customization policy, system implementers and developers should contact Oracle Support Services for clarification.

That said, if tools such as Oracle Forms, JDeveloper for Oracle Applications, Oracle Workflow Builder, and others that we use to build customizations do not behave as documented, we are entitled to address an issue with Oracle Support and raise a support call. The best course of action is to create a very simple test case that is not dependent on our custom code but is of generic nature. As we said earlier, all the documented features of Oracle tools and Oracle Applications are fully supported and will be dealt with by Oracle Support.

Concept of E-Business Suite Environments

In organizations that implement or already have implemented E-Business Suite, you will find multiple copies of Oracle E-Business Suite installations in use. The installations can be either on the same machine or on different physical machines. Each such installation is called an instance or an environment of Oracle E-Business Suite and consists of E-Business Suite software, an Oracle database including the data files, and Oracle server software.

An instance has a unique purpose; for example, if the customer is already running their business operations on E-Business Suite, they will always have a production instance. An E-Business Suite developer should never directly make any code changes to the production environment. The code changes must be first done to a development environment, and from there on promoted to test, and finally to production systems. The promotion of code changes must be scripted where possible to avoid human error.

An E-Business Suite developer engaged in the task of extending or customizing a module within an E-Business Suite at a customer site will typically find that customer is either in the implementation or production phase. The environments that support the implementation process are different from those required to support a post “go-live”

running production instance of E-Business Suite. For instance, during the implementation phase, it is usually required to perform a master system configuration; develop and perform system testing of interfaces, conversions, and customizations; test the performance of the final system and infrastructure design; perform a UAT (User Acceptance Test); and train the end users and go live with the production system. Obviously, the production system requires fewer environments. Customers that are already running "live" production systems need support and development environments for the future system enhancements. They also need to test patches that fix production issues and a separate UAT environment for the final sign-off prior to applying changes to the production environment.

When it comes down to detail, every implementation is different in terms of used number and types of environments that support either the implementation process or live production system. Here is a brief description of some of the environments that exist during an Oracle E-Business Suite implementation process:

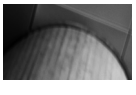
- **Master environment** This environment is used for the main configuration setup of the system. Although it does not contain any transactional data, it is important that the master environment is managed by a very strict change control as this environment contains production (master) setup.
- **Development environment** This is where developers design and build extensions and customizations. The developers are usually granted very powerful access rights for both E-Business Suite and the operating system that hosts the system. For instance, the developers may be granted System Administrator or Application Developer responsibilities.
- **Testing environment (also known as UAT)** Developers usually do not have an APPS database schema password to this environment. This is where users sign off on customizations and configuration changes.
- **Deployment environment** Once the users have finished their User Acceptance Testing on a UAT instance, patches/scripts can then be promoted to a Deployment instance for final checks. Effectively, applying patches on a Deployment instance is a dry run before applying code changes to a Production instance.
- **Patching environment** Oracle delivers their code changes, bug fixes, and product updates through *patches*. The patches can be downloaded from the Oracle Support website and applied by E-Business Suite database administrators (Apps DBAs). Apps DBAs can use the patching environment to perform sanity checks for patches delivered by Oracle.

- **Support environment** If a user reports an issue on the production system, it is a good idea to reproduce the problem on a copy of the production system. Such copied instances are referred to as *clones*. The support environment is exclusive to the support staff, where developers do not make changes directly. This environment is usually the most frequently cloned environment in cases where E-Business Suite implementation is running a live production instance. Frequent cloning helps the E-Business Suite support staff to reproduce production issues.
- **CRP environment** The conference room pilot environment is where someone, usually an implementation team, gets buy-in to their product offering from the wider user and business community during an implementation. This environment is usually used for sign-off during new implementations.
- **Migration environment** For new implementations of Oracle Applications, developers are tasked with migration of data from the old legacy systems into Oracle E-Business Suite. This is where repeated data migration can take place before the migration code gets frozen and ready for user testing.
- **Production environment** This is where the business runs its day-to-day operations.

Generally, E-Business Suite technical developers shouldn't be too concerned about the variety of environments, as their focus is predominantly concentrated on the development environment. In very simple terms, the life cycle of extensions and customizations could be summarized as follows: the developer performs the development and unit testing in the development environment, and the code gets promoted to the testing environment. Following successful testing, the changes are applied to the production environment.

There can be more than one development environment for any implementation as well as a live site, especially when some of the bigger modules are being implemented with different timelines. Nevertheless, the changes in each development environment should ideally be tested on a common test (UAT) environment.

The changes in the development environment must be scripted in all cases where possible. As a rule of thumb, everything except for functional configuration can be scripted. To promote functional setup and configuration, the implementers of E-Business Suite can use the iSetup module, which is used to promote functional changes between various E-Business Suite environments.



NOTE
The process of automating of the code delivery helps avoid human errors; changes can quickly be promoted to other test instances, and this approach also ensures a tight control over the changes that affect production instances.

Concept of Common Entities and Common Data

Sometimes people refer to the common data as *shared entities*, but you can also think of them as business objects or entities that are common to a number of different business functions. For example, entities such as customers and suppliers can be referenced by multiple Oracle Applications modules.

You may have heard that Oracle Applications are built around a single data model, which, in essence, means that within a single database you can find a single definition of your customers, suppliers, employees, inventory items, products, and all the other important aspects of a business or an organization. In contrast to this idea of a single data model, organizations tend to build or implement new applications to meet their business needs as they grow, ending up with "point-to-point" solutions between the systems because new applications need to share the existing data with other applications in the organization. As the systems alongside the business continue to grow, the number of interfaces between disparate applications will also grow. For example, Human Resources–related data about employees could be stored in one database, while financial data is stored in another system.

Figure 1-1 represents such systems, where the applications are added one after another as the business needs grow, and as a result, end-to-end interfacing between them starts to look incomprehensible. It is perfectly possible to make such applications collaborate to connect different business processes; however, when major changes occur in one application, it will start having a domino effect on other components of the system and make it more expensive to maintain. Oracle E-Business Suite is trying to address this issue by integrating around a single common data model. The idea of this model is to allow us to create and maintain a single common business definition of employees, students, customers, suppliers, products, and other aspects of a business or an organization, so everyone in that organization has an instant access to the common data shared by different applications. All the applications collaborate with each other, share the same information, and can be run in one global installation of a single database. Oracle E-Business Suite is designed and shipped as a preintegrated set of applications, but organizations and businesses are free to implement a single application, multiple applications, or all of the applications that comprise Oracle E-Business Suite. This modular approach is a key integration enabler that allows us to integrate with already existing applications.

It is important for developers to keep this in mind, as almost all of the custom development efforts in Oracle Applications will reference the common or shared entities. In addition, they are not documented in a single user or implementation guide as a part of the Oracle Applications documentation library. If you search the Oracle Applications documentation library online or Metalink (Oracle's support services website), you'll see that common entities are referenced in different

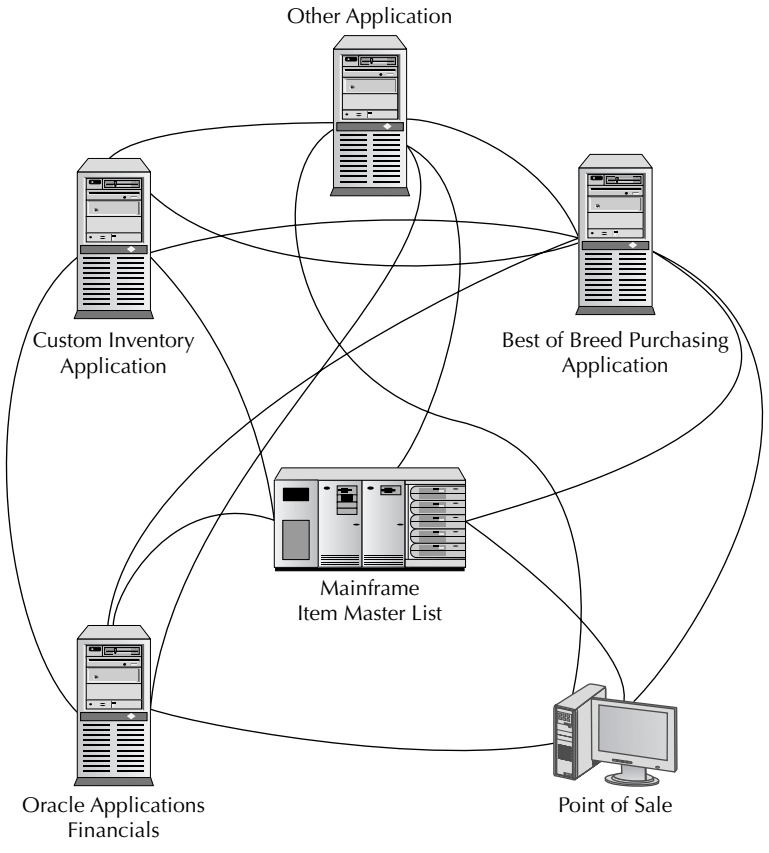


FIGURE 1-1. *Fragmented point-to-point interface model*

implementation and user guides for the multiple products that your organization has implemented. If you think of the common data as business objects shared and referenced by multiple modules, as indicated in Figure 1-2, you can then say that the common data represents reusable entities defined as a one-time exercise in one product and then shared and reused by other applications. For example, suppliers defined in Oracle Payables are shared between Payables, Assets, and Purchasing applications. Similarly, items defined in Oracle Inventory are shared by Purchasing, Order Management, and Receivables. Further examples of the shared entities are Organizations, Locations, Employees, Units of Measure, and Items.

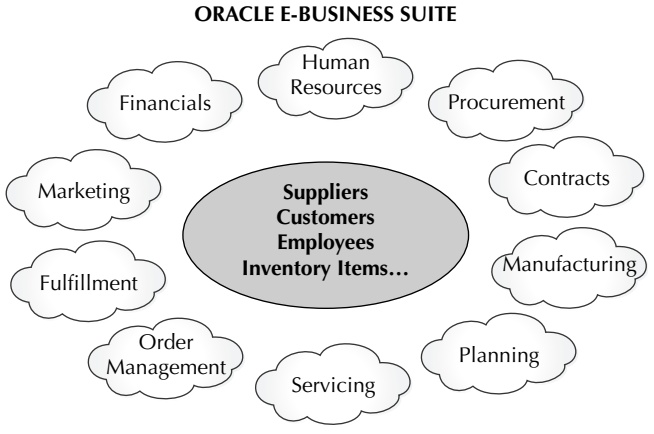


FIGURE 1-2. *Shared data model*

Examples of Common Entities

Figure 1-3 gives a simplified view of an example of data sharing between the different products in Oracle Applications. In the figure there are three common entities that are shared across the modules: Items, Customers, and Suppliers.

- **Items** Order Management and Purchasing and many other Oracle applications use the definitions of items configured in Oracle Inventory. Items are usually the things that an organization or a company makes, purchases, or sells. Different applications use items for different purposes. In Oracle Inventory, items are used for stocking process, planning, and cost; in Payables, items are used in supplier invoices; in Receivables, they are used as units to bill the customers.
- **Customers** In Figure 1-3, the customer purchase orders are created in Order Management (OM). Sales orders define what products are shipped to the customers. After shipping the products to your customers, you invoice the customers through Receivables, and Oracle Inventory adjusts the quantity of the products currently held. The customers created in Order Management are shared with Receivables and vice versa.
- **Suppliers** In the example, the suppliers are defined through the Oracle Purchasing module. The suppliers are business or trading partners that deliver goods or services of some kind. The supplier invoices are entered into the Oracle Payables and matched to the purchase orders in the Oracle Purchasing module. You can create suppliers in different modules such as Payables or Purchasing.

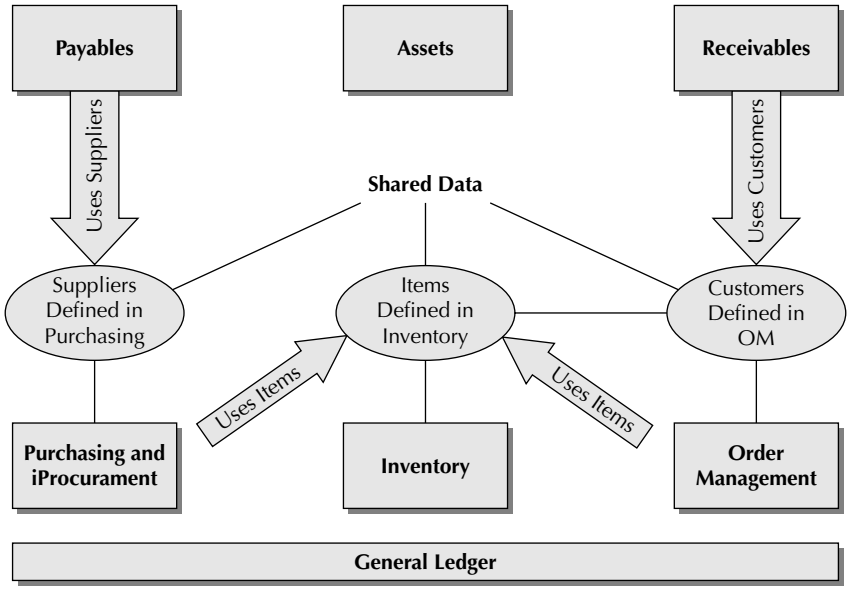


FIGURE 1-3. *Example of data sharing*

Summary

In this chapter, we introduced customizations, personalizations, E-Business Suite instances, and the concept of data sharing. However, this is by no means a complete list of topics that developers should keep in mind when venturing into building custom code and extensions. We also highlighted the difference between system configuration, personalization, and customization. It is important to understand that the custom logic in the code that we develop to extend product functionality is not supported by Oracle but by ourselves. However, the tools and various technology frameworks used with E-Business Suite are supported by Oracle Support, and any issues with them are usually dealt with promptly by Oracle Support staff.