

CHAPTER

1

A Fusion Applications Product Overview

2 Managing Oracle Fusion Applications



It's probable that no one person can completely understand the whole of Fusion Applications—not because of its complexity or a lack of information, but simply because of its size and scope. The broad range of easy-to-use functional features support all core business operations, while behind the scenes it runs a complex platform of technical components, each with intricate setup, function, and execution. Fusion Applications contains hundreds of functional features within each product area, and the first release of Fusion Applications comes with more than 80 product areas.

(Hundreds of features × 80 products) × (Platform complexity) = You get the picture!

Although this overview covers all the key areas that are essential for an appreciation of Fusion Applications, there remains a strong emphasis on leveraging additional chapters and reference material for specific detail. To cover even the foundation-level information, without running into thousands of pages, this high-level discussion is divided into four sections and two chapters.

This chapter is focused on the finished product itself—what kinds of things it can do from a business and end user perspective—and is divided into two main sections:

- **Business architecture** The business operations supported by Fusion Applications.
- **Functional architecture** How the application hangs together.

Chapter 2 takes the discussion a step deeper, providing an overview of the technology components that run underneath the Fusion Application product features, since this is essentially the responsibility of the Application Manager. This chapter included the following two sections:

- **Technical architecture** The technologies at work and the complete technology stack.
- **Fusion Applications ecosystem** Additional supporting frameworks and components.

To explain the purpose of each component, this discussion starts with the business-user details and drills down through the supporting technologies underneath.

Business Architecture

A recent cultural shift is at the heart of Fusion Applications: the emphasis on technology as a real business enabler. As organizations continue to tighten their IT spending, all investments must provide significant (and immediate) benefits to the bottom line. Return on investment (ROI) really matters now; where once technology was viewed as having the potential for improvement and competitive advantage, this must now be clear, and, most importantly, it must be proven.

So how has this shaped Fusion Applications? The first step was to review the best-in-class features of each of the existing Oracle Application products, those that many businesses were already using to ensure their own success. Next was extensive requirements gathering, based on the needs and desires collected from literally hundreds of organizations. Subsequently, everything was rigorously designed with a core focus on enhancing simplicity, completeness, and flexibility. These factors ensure that Fusion Applications is more focused on supporting real work processes than any of its predecessors.

The Business Process Model

One simple goal runs through the whole of Fusion Applications: to support a comprehensive list of best practice business operations. This list was built from several sources, including industry experts, Oracle's own Application product features, as well as detailed consultation with some of the world's most successful organizations. This analysis captured the precise way in which many organizations actually work, from boardroom to shop floor. Two simple examples of business process flows (actually known as *composite flows*) are the processes and actions involved between the purchasing of raw materials to analyzing subsequent supplier payments—the procure-to-pay flow—and the process that starts with a customer sale and ends with being paid for the goods—the order-to-cash flow. Clearly, the closer the application features are aligned to what business users actually do, the more successful its adoption will be. This set of best practices upon which Fusion Applications features are based is known as the *Business Process Model (BPM)*.

The BPM is leveraged in several ways in Fusion Applications. This includes tying code components to functional activities (via *Logical Business Objects*), organizing the user interface flow (via *taskflows* and

4 Managing Oracle Fusion Applications

activity guides), and ensuring that all the supporting documentation and online help is provided in the right place and at the right time.

The BPM structure is based on a hierarchy of nodes that starts by identifying the industry context and then decomposes it into four child flows, as shown in Figure 1-1.

Starting with Level 0, Industry, each level provides a finer level of detail than its predecessor, with Level 4, Task, being the precise actions a business user would perform in his or her job. The BPM content remains “application agnostic” at Levels 0 to 3, with only Level 4 providing details that are specific to Fusion Applications. An example of this hierarchy is the Level 1 Business Area of *Procurement* that has 12 child Level 2, Business Processes, one being *Manage Requisitions*. *Manage Requisitions* itself has four child Level 3 Activities, such as *Approve Requisition*, which in turn has seven Level 4 Tasks, including, for example, *Reject Requisition Request*. The BPM architecture

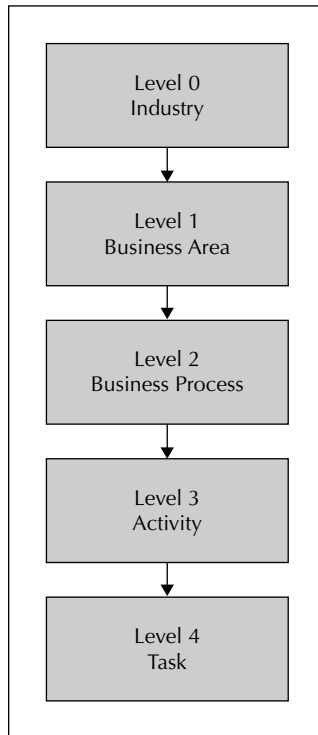


FIGURE 1-1. *The BPM hierarchy*

forms an outline for all Fusion Application functional features, and with around 11,000 tasks, it covers all core business operations and many additional operations.



NOTE

Version 1.0 of Fusion Applications focuses on common core flows that apply across many different industries. Therefore, the Level 0 contains essentially just one cross-industry root node. Further, more 0 levels are planned for forthcoming releases.

As well as being a design architecture, the BPM (unsurprisingly) has its origins firmly in Business Process Engineering. This allows organizations to map the capabilities of Fusion Applications to their own internal business processes and procedures as part of requirements analysis. Taking this a step further, using the standards-based tools available (such as Oracle's Business Process Management Suite) and the modular nature of the Fusion Applications business processes, you'll find it easier than ever before to perform setup, configuration, and even extension to meet those needs. (The book's appendix contains further reading on BPM and Business Process Engineering.)

Fusion Applications 1.0 Features

Even in the first release, Fusion Applications includes a broad set of features that support the majority of day-to-day business operations. As discussed, product features are arranged primarily by BPM; however, for simplicity and clarity, all features are also categorized into a high-level product grouping, known as a *product family*. Each product family may contain part or all of a business process—for example the Procure-to-Pay business process is mostly part of the Procurement product family, but the final stages of the flow around invoice reconciliation and payment fall into the Financials product family. An overview of features in each of the six product families in Fusion Applications 1.0 is discussed next. You'll find more information in the Appendix.

Financial Management

This product family represents a full suite of accounting and financial management tools that will satisfy most private sector business needs

6 Managing Oracle Fusion Applications

(with public sector and budgetary control features anticipated later). It includes General Ledger, Accounts Payable, Accounts Receivable, Asset Management, Payments and Collections, Cash and Expense Management, and a range of supporting common modules and reporting capabilities. It also natively supports many key legislative requirements of the major geographical territories (such as USA, Western Europe, Asia, and Latin America), although full localization support will be developed in later releases. In addition, Fusion Financial Management is immediately ready for integration with Oracle's Hyperion product range, offering exceptionally powerful financial performance analysis.

Project Portfolio Management (PPM)

This product family includes full support for the following: Project Costing, Project Billing, Project Control, Project Performance Reporting, Project Integration Gateway, and a range of integrated analysis reports. This family focuses on private sector businesses and does not include the budgetary control pieces in version 1.0. One key benefit of Fusion PPM is that it has native integration with Oracle Primavera, offering powerful features to support projects in many vertical industries such as engineering and construction, utilities, and high-tech.

Human Capital Management (HCM)

The HCM offering covers all the core features required by modern enterprises, including Human Resources, Global Payroll, Workforce Service Delivery, Total Compensation, Workforce Management, and Talent Management. Each of these comes with embedded performance indicators and powerful reporting capabilities. As with the other features, this is primarily focused around private sector use in version 1.0.

Supply Chain Management (SCM)

This product family offers the following product features: Product Information Management, Asset Management, Global Order Promising, Cost Management, Distributed Order Orchestration, and Logistics (that includes Shipping, Receiving, and Inventory). Fusion Applications version 1.0 does not yet include full manufacturing support, such as Materials Requirements Planning (MRP), Quality, or Work-in-Progress. Arguably, the most exciting offering for SCM is in its Distributed Order Orchestration (DOO) product that provides a new central

solution for the management and fulfillment of sales orders. Leveraging the latest technologies, DOO offers unprecedented levels of flexibility that should meet the needs of almost any organization.

Procurement

The Fusion Procurement product family offers a wide range of features that help make the corporate buying process much more efficient and effective. This includes features for Purchasing, Self Service Procurement, Sourcing, Procurement Contracts, Supplier Portal, and Supplier Model. It also includes comprehensive performance management and spend analysis reporting.

Customer Relationship Management (CRM)

The CRM product family offers a broad range of features, specifically based around Sales and Marketing activities for version 1.0. This includes Customer Master, Sales, Marketing, Mobile, Outlook Integration, and the same detailed reporting capabilities. In addition, the CRM product family hosts the complimentary Incentive Compensation product and features.

Governance, Risk, and Compliance (GRC)

To satisfy the need for business operation control and recent regulatory requirements, this product family offers the following features that are fully integrated with the other product families: Financial Compliance, IT Risk and Compliance, Issue Manager, Risk Manager, Access Controls, Transaction Controls, and Configuration Controls.

Functional Architecture

Many concepts and features are included in this section—they all simply provide the flexibility to implement Fusion Applications in a way that best suits an organization. These common datasets and internal business structures form the foundation upon which transactions are processed, features operate, and performance is monitored.

Functional Architecture Objects

All Enterprise Applications contain objects that are defined once and then reused in many places. Common examples are employee data such as e-mail addresses and job levels, and organizational structures such as departments,

8 Managing Oracle Fusion Applications

divisions, and addresses. The following six concepts will provide a basic understanding of the core functional structures within Fusion Applications. As we'll discuss later on, truly effective Application Management requires a strong appreciation of all the underpinnings of the application, both technical and functional.

Trading Community Architecture

Transaction-related data is usually denormalized in enterprise applications, so storage is optimized and any changes are made only once. Examples include the definitions for suppliers, customers, people, locations, and contact details. So that every Fusion Applications feature can access these definitions in a uniform and consistent manner, a central data model is used to help optimize designs, features, and even system performance. This model, the Trading Community Architecture (TCA), is an evolution of the same model that existed in E-Business Suite. As demonstrated in Figure 1-2, the TCA model represents physical objects such as people and companies as *parties*, and allows each independent party to have one or more *relationships* among themselves and

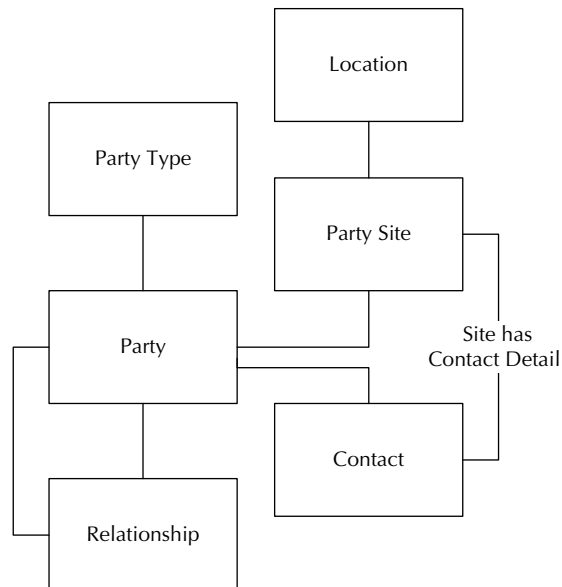


FIGURE 1-2. A section of the Trading Community Architecture

other parties. An example is the employee relationship between a person party and a company party. A party can also have one or more locations (addresses and some contact details) recorded as *party sites*.

This standardized data model is leveraged right across Fusion Applications and forms a vital ingredient to ensure the integrity of the basic functional data. Although all of this will remain essentially invisible to the business user, the Applications Administrator will occasionally benefit from a basic understanding of the model. Potential uses include analyzing issues around functional setup, assessing extended reporting needs, and managing database performance.

Enterprise Structures

All businesses have internal subdivisions, which may be geographical, such as international or regional departments, or delineations related to work specializations, such as purchasing, production, and sales. All Enterprise Applications need to mirror these internal structures to ensure that they accurately represent and support each part of the organization. Fusion Applications has taken the best such capabilities from the existing Application products, added some new pieces, and delivered its own method for setting up what are known as *Enterprise Structures*.

To emulate the organizational structure inside the application, *business units* are used to represent logical or physical divisions. Business units roll-up into parent *Enterprise* business groups. Business units have two main purposes: they are used for the grouping of data, functions, and transactions for better management and reporting, and they are a method by which common reference data can be shared.

Here's a simple example: The California Business Unit provides centralized procurement business functions (that is, a purchasing department) and associated item data for five other West Coast business units, with the Florida Business Unit providing the same for five East Coast business units. In addition, the Colorado Business Unit may provide all invoice payment and accounting functions for the whole organization. In this example, orders can be reported based on each of the ten separate business units that raise them, the two business units that provide the centralized purchasing function, or even just the Colorado Business Unit that manages the centralized finance. This all helps improve administrative processing (especially for shared service centers), and it supports all performance analysis and legislative reporting needs.

10 Managing Oracle Fusion Applications

In addition to Business Units, Fusion Applications contains a method for creating *legal entities*. This allows separate legal reporting (such as accounting, fiscal, and tax) for each discrete finance and employment structure that exists, and it is most commonly used between different countries.

Effective Dates

Fusion Applications is designed to flex with an organization, and as organizational changes occur, the application can be adjusted accordingly. Adjustments are commonly required to the active reference data within the application—for example, from active order attributes to employee's job codes.

Date Effectivity is the mechanism by which any reference data can be quickly made inactive or active. In addition to making immediate changes, data can also be scheduled for use on future dates as well. Retroactively, the Date Effectivity feature ensures that all historic transactions retain their original reference data, improving performance reporting and audit capabilities, as well as helping to eliminate potential mistakes.

SetID

Although the Date Effectivity feature allow organizations to version-control their reference data, SetID simply acts as the container in which reference data can be created. Each separate set of reference data is associated with one or more specific Enterprise Structures (also illustrating how these functional architecture pieces work together). Since this data needs to be defined only once, no duplication or associated maintenance overhead occurs. Using our earlier example, the California and Florida business units that perform buying functions might be assigned one SetID for Project Expenditure Types, allowing them to assign customer orders to the appropriate internal projects. A different Project Expenditure SetID might then be used for other business units that specialize in the fulfillment of internal orders only.

Tree Management

Fusion Applications users can not only set up reference data, but they can also organize it into custom structures that represent its internal relationships. Tree Management allows you to create tree-like hierarchies, first by registering a *Tree Structure* and then associating data values to it. A simple example would be the job roles and teams within an organization.

Tree Management offers several key benefits, many of which are also applicable to other Functional Architecture components. Although trees are exceptionally flexible, they remain independent of any specific implementation, allowing the same tree to be reused in many different places. This reduces the maintenance overhead resulting from data changes. Another key benefit of having data structures represented and stored in a consistent way is that they're readily available to the advanced analysis and reporting capabilities prevalent in Fusion Applications. Although there are some other advantages, one final simple point is that trees represent a very powerful visual metaphor. This is demonstrated in Figure 1-3, where the page allows the business user to understand immediately the relationships involved, while ensuring accuracy and security at all times.

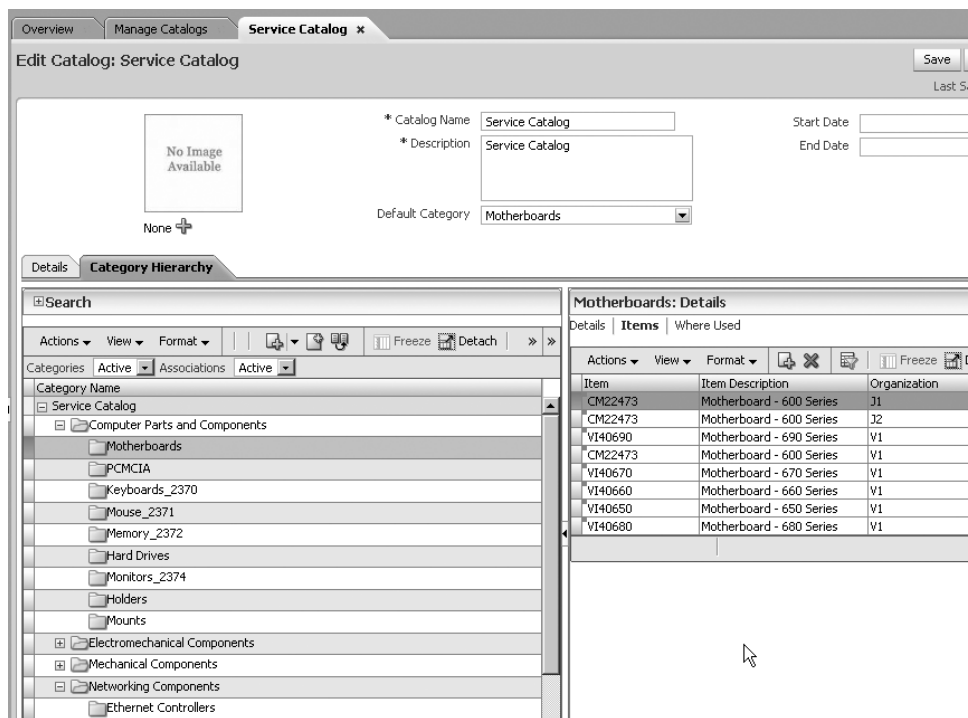


FIGURE 1-3. Tree structure shown in the Item category

12 Managing Oracle Fusion Applications

Approval Management

Business transaction approval is a compulsory feature for Enterprise Applications, and Oracle Applications successfully used a workflow-based system for many years. Both E-Business Suite release 12 and PeopleSoft version 8 also include advancements that provide the implementation-independent setup and execution of a rule-based approval system.

Fusion Applications has taken this concept even further by leveraging the Approval Management Service (AMX) that exists as part of the Human Workflow Business Process Execution Language (BPEL) capability inside Oracle Fusion Middleware. This again abstracts many pieces of the approval process and dedicates best-of-breed tools to complete specific tasks, such as building the approvers list, sending alert notifications, and processing responses.

Functional Setup Manager

Although this book does not delve into the world of application installation and implementation (for many reasons), and although detailed resources are included in the Appendix, one setup-related piece is definitely worth discussing here. With its origins and intentions similar to that of E-Business Suite's iSetup product, Functional Setup Manager is a self-contained feature that provides a platform upon which the implementation can occur. Centralizing the functional setup in a single integrated tool helps eliminate substantial project complexity and allows actions to be planned, executed, and reported, all within the context of the Fusion Application instance itself. That's not to say it totally replaces full-scale Enterprise Application project planning; however, within its own limited scope it is simple, flexible, and comprehensive.

Functional Setup Manager has two main capabilities. The first is the mechanism to define an *Implementation Project* and then choose the desired products and features that will be implemented (known as *offerings*, *features*, and *options*). As shown in Figure 1-4, this project will then automatically contain a sequenced list of all the actions required to complete the related setup (in a *task list*), automatically factoring in any dependencies that may exist.

Each task can have additional information and documents associated with it, which, together with completion tracking and team collaboration features, provides powerful management capabilities. Projects can also be extended to include any additional organization-specific tasks required, such as extra integration work. For applications administrators, it's immediately obvious how this will prove to be invaluable, representing a useful tool for the management and control of functional setup data.

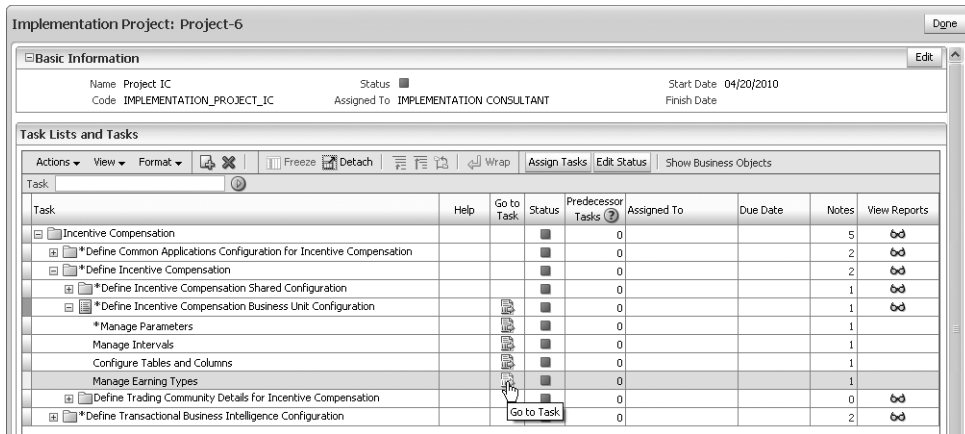


FIGURE 1-4. Functional Setup Manager showing tasks within a project

The second main feature of Functional Setup Manager is its ability to import and export functional setup data. Export and import files are based on a set of standardized templates, which themselves can be part of a *Configuration Package* to allow further extension and customization. While hugely beneficial for implementation, this feature is also useful for application administrators who are responsible for many separate instances; at the press of a button, some specific setup values (or everything) can be made available and quickly imported into another instance ready for immediate use. This marks a significant improvement on traditional methods such as unwieldy database dumps and complex custom extract scripts.

Finally, Functional Setup Manager has an associated tool that provides some technical implementation services. The *Topology Manager* acts as an internal store for data that links the logical separation of implementation and setup (pillars, offerings, options, features, and so on) with the physical deployment details, such as business objects and end points for services and applications. This represents more of a back-end repository, and although it's helpful to be aware of its existence and function, it's not really a systems management tool, despite its name.

