CHAPTER 14

Oracle Projects
The scope of information management for modern corporations is growing from the accounting core. Oracle Projects has been designed for a huge variety of project-oriented businesses. Key performance indicators such as profit margin are often measured by product, customer, or customer type and rolled up into division, region, and company. The project dimension allows a company to do the same by contract. It might sell a service such as facilities management or consulting. It might sell infrastructure setup such as optical-fiber laying. Projects also might not be billable to clients but rather be a way of controlling the cost of a program, perhaps retraining staff after a technology change or managing the commissioning of a new production facility.

The key dimension is time; a project is time bounded. It has an objective or set of objectives to which a value and usually a budget have been attached. The goals of using project costing and billing software are to

- monitor the costs against budget quickly enough that remedial action can be taken if necessary,
- account for direct and indirect costs,
- show the profit margin accruing from the project, and
- show costs and revenues accumulated to date (other subledgers and the General Ledger are limited to reporting by accounting period).

Enterprise project management involves the collection and coordination of corporate resources (such as people, money, and hard assets) to accomplish a predefined scope of work in a scheduled time frame and budget.

The Oracle Projects suite enables project managers to effectively oversee their projects, assess progress against predetermined milestones and budgets, staff their projects with appropriate talent, and quickly generate a wide variety of reports. It also helps virtual and globally distributed project teams to efficiently communicate, collaborate, and complete tasks in time. Oracle Projects also gives corporate executives the ability to quickly see how projects are performing across the enterprise.

Oracle has made a big investment in the Oracle Projects suite, combining all the needs into a set of products that can enable a company to track and manage its projects from different angles. Some of the modules that are part of the Projects suite are

- **Oracle Project Costing**  Provides an integrated cost-management solution for all projects and activities within an enterprise. Run across multiple currencies and organizations, it acts as a central repository of project plans and transactions, processes project costs, and creates corresponding accounting entries to satisfy corporate-finance requirements.

- **Oracle Project Billing**  Enables enterprises to simplify customer invoicing, streamline corporate cash flow, and measure the profitability of contract projects. It provides features to review project invoices online and analyze project profitability and corporate impact of project work.

- **Oracle Project Resource Management**  Empowers an enterprise to make better use of the single most critical asset: people. It enables efficient coordination of project resource needs, profitability, and organization utilization through the location and deployment of qualified resources to projects across the enterprise. It is integrated with Oracle HRMS to efficiently deploy human resources in various projects across enterprise.
Chapter 14: Oracle Projects

- **Oracle Project Management** Enables integrated project planning, tracking, and real-time project-performance management. With Oracle Project Management, project managers can proactively plan and forecast their projects, manage change and performance in real time, focus on desired project outcomes rather than data management, and make better decisions with less effort.

- **Oracle Project Collaboration** Provides a secure and intuitive user interface through which a team member of a project can see assigned tasks, issues, deliverables, and other project-related information. It has structured workspaces such as the Team Member home page, with the help of which team members can work together more efficiently, make more effective decisions, and deliver superior results faster.

- **Oracle Project Portfolio Analysis** Leverages the rich project-management functionality of Oracle Projects to facilitate evaluation and collection of projects in a portfolio. It uses financial criteria, strategic goals, and information on available funds to help you evaluate, prioritize, and select the right projects to match your business objectives. It provides a full range of portfolio-analysis reports, charts, and graphs.

The Projects suite integrates with the other modules by sharing tables and through open interfaces. Shared information includes Human Resources organizations, employees, and Account Receivables customers. Cross-module exchanges include one concurrent process to populate the Oracle General Ledger interface tables with costs and another for revenue. Costs, which come primarily from Oracle Purchasing and Accounts Payable, can be recognized as commitments and obligations even before they are invoiced. Revenues can be recognized as they are earned, which can be earlier than they are billed. Costs that have been marked as project specific come into the module through the cost-transaction import interface.

**Setup**

Effort expended in the setup phase is well rewarded when the resulting business processes are straightforward and simple to operate. Ensure that you settle as many business requirements as possible early in the implementation, so that the subsequent test cycles have a solid base on which to start. The setup decisions you make determine the choices you have for the accounting and reporting. *AutoAccounting* is the automatic account-generation engine used by Oracle Projects to determine accounts for transactions that are imported into the module without accounts already attached and for transactions that originate from the module itself. In Release 12, with the introduction of Subledger Accounting (SLA), there is another level of accounting definitions that can overwrite the definitions of AutoAccounting, providing greater flexibility (see Chapter 4 for more details). The standard reporting tool for the module is the Project Status Inquiry screen. Both of these features are described in greater detail later in the chapter.

**Organization Setup**

Organization setup is one of the keystones of the E-Business Suite; Oracle Projects relies heavily on a carefully set-up organization structure. Organizations must be planned at a global level. They play a significant role in Oracle Human Resources, Oracle Inventory, Oracle Manufacturing, and several other modules. Oracle Projects uses the following types of organization classifications:

- business group,
- human-resources organization,
operating unit,
- project/task-owning organization,
- project expenditure/event organization, and
- project invoice-collection organization.

The business group is the top of the organizational tree. There is a seeded Setup Business Group, which is best to use as the basis for the first real business group. There often is just one business group that represents the holding company. It holds the legislative rules (such as tax references) and terms of employment (such as default working hours) for employees.

For Oracle Projects, the business group is the owner of the *project-burdening* organization hierarchy, which defaults the *burdening multipliers* for an organization. Burdening is the process of allocating overhead costs to projects—for example, stationery or administrative office costs. One key constraint needs to be understood when designing the business-group structure: employees in one business group cannot charge cost items to a project owned by another. So in the case where employees charge their time to projects for other companies in the group, those companies need to be in the same business group. Employees are assigned to work in human-resources organizations. These can be one and the same as the project-owning organizations (which themselves may also be operating units) so that labor can be charged to projects without the need to set up cross charging.

Organizations can also be set up to own projects and tasks. This relationship can be used to drive the accounting (via AutoAccounting) and reporting. It is assigned to an operating unit (and therefore the user’s Responsibility) through the organization-hierarchy branch. The introduction of Multi-Org Access Control (MOAC) in Release 12, enables companies that have implemented a Shared Service operating model to efficiently process business transactions for an unlimited number of operating units within a single application responsibility. By default, users can see a project only if they are set up as a key member of it. Project expenditure/event organizations can incur expenditure and create revenue events for a project. AutoAccounting can also use these organizations to derive account codes.

It is important to settle the organization structure across the whole implementation rather than on a module-by-module basis—it is a cross-functional exercise because organizations exist independently of their types. The organization setup resides in Oracle HR tables. These are accessible either through Oracle Human Resources (if it is installed) or through forms in the other modules if the install is set to “shared.”

No license is required to use the HR shared tables when the other modules are installed. An organization is set up in the HR_ORGANIZATION_UNITS table regardless of which module’s organization-setup screens are used, and organizations set up in one module are viewable in another, so that organization types can be added as shown in Figure 14-1. For example, rather than have two organizations set up—one by the Inventory setup team, the other as a project owner by the Projects setup team—which actually are the same business entity, set one up with both organization types assigned.

Once the organizations themselves have been created, the hierarchies can be put in place. This is important for Projects because it defines the relationships between business groups, operating units, and the project-specific organizations. For example, it is used to determine which projects a user can see from a Responsibility (which is linked to an operating unit). As with organizations, hierarchies are an HR concept that lives independently of any one module; thus they can be set up in either the Projects module or the HR module.
Oracle Projects uses employees as key members (such as project manager or quantity surveyor) and associates them with costs, which also need to be set up. There is an employee-and-assignment setup screen provided in the module in case HR is not installed. This version of the employee setup screen is more comprehensive than that found, for example, in Payables, because it can keep a history of the employees’ assignments, which is significant for reporting. Employees are referenced in straight-time and overtime cost transactions; historic reporting of these could derive the position and job the employee held at the time.

**Project Setup**

A project can have one of three different type classes: capital, indirect, or contract. Project types can be user defined but must be associated with one of the three type classes.

A capital project is one in which the objective is to build something that will be capitalized as an asset. Oracle Projects is used to track your construction-in-process assets until you capitalize them by placing them in service, when they are interfaced to Oracle Assets. A project can result in one or more capital assets. These are defined at project level and then can be (but do not have to be) associated with tasks by using the grouping level. They can be associated with the whole project or with tasks at any level of the work-breakdown structure, so that expenditure items from many tasks will be included in the capital cost of a single asset when it is placed in service.

An indirect project is one to which overheads are assigned (which can subsequently be allocated to a number of other projects)—for example, a project office or depot. For example, in a
consulting organization, an indirect project can be used to monitor sick leave, training, holidays, and other nonbillable activities. Each of these activities would be created as a task on the indirect project. In a construction business, the administrative function can be an indirect project. The allocations feature enables the costs associated with indirect projects to be allocated across projects of different types.

A contract project is one that is run to be billed to a client. Billing can be on many different models—for example, time and materials, cost plus, or fixed price. A contract project must be funded by a customer agreement. There might be situations in which it is expedient to set up several projects for a customer, in which case a single agreement can cover all of them. For example, a customer might want to have a hospital built and then run for him or her. These might be best reported on as two separate projects; however, the customer has a single contract for funding purposes, which will be shown in Projects as an agreement. It is mandatory to create a revenue budget, and until it is baselined, no revenue can be generated on the project. If you use a project template with an associated agreement template to create a new project, the system creates a Quick Agreement for you.

A contract project can be owned by one or more customers, either sharing a single agreement or having separate ones. Revenue can accrue either at the top task level or at the project level. However, if it is accrued at project level it is calculated at this level, too; therefore, the detail will not be available for reporting. The key project attributes and their meanings or uses are shown in Table 14-1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning/Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project-owning organization</td>
<td>Organization against which revenue and expenditures are tracked.</td>
</tr>
<tr>
<td>Project type</td>
<td>Links to project-type class (indirect, capital, or contract); holds the costing, capitalization, and billing methods.</td>
</tr>
<tr>
<td>Status</td>
<td>User-defined statuses must be associated with one of six project-system statuses.</td>
</tr>
<tr>
<td>Start and completion dates</td>
<td>Period during which expenditures can be charged to a project.</td>
</tr>
<tr>
<td>Key members</td>
<td>Identified by role type, such as project manager or quantity surveyor; links to employee and identifies who can make changes to this project.</td>
</tr>
<tr>
<td>Customers</td>
<td>Links to customers set up in either Projects or Receivables.</td>
</tr>
<tr>
<td>Project classifications</td>
<td>Two-layer structure: class category and classification; used for reporting—for instance, for the category of “Public,” classifications might be “Transport” and “Other”; can also used to determine accounting.</td>
</tr>
</tbody>
</table>

TABLE 14-1. Key Project Attributes
Task-Setup Attributes

Tasks are arranged into a multilevel hierarchical work-breakdown structure that is unlimited both in the number of levels and in the number of tasks. In practical terms, the depth of the structure is limited by reporting constraints: there are only 20 characters provided for reporting the fully qualified task number, including delimiters. Therefore, a construction project might have three top tasks of preliminaries, main building work, and retention-period activities. Under these there might be some middle-level tasks that group together the lowest-level tasks. It is these lowest-level tasks against which actual costs are accumulated. Revenue and billing only apply to the highest-level tasks.

The work-breakdown structure is a simple hierarchical list of tasks that make up the project. The top level of the structure is usually used to show the phases of the project, such as preliminaries, design, construction, testing, and commissioning. The lowest level is the level at which costs are collected. Therefore, this should be the level at which costs are realistically going to be managed.

Standardization in task numbering will help make project reporting more easily readable, because tasks will appear in alphanumeric order in the Project Status Inquiry screen and in standard reports. Task numbers such as 01 and 02 for top tasks, with 01.1, 01.2, 02.1, 02.2 as their subordinates, will add clarity. There is a 20-character task name and a 250-character description, so numbering tasks in this way does not have a detrimental effect on the richness of the information held. Note that task numbering might have an effect all the way through the business, in the case where project and task setup is pulled through from a third-party system; the naming and numbering conventions will have to change in the originating system.

Quick Entry Screens and Templates

Although a company might have only very few projects, the possible number in the software is unlimited; thus, efforts have been made to make the data entry as short as possible. Projects are entered using the Quick Entry screens, which derive much of the information from the project type and the template used. Projects can be created from another project or a template. The Quick Entry screens can be configured in the module-setup stage to suit the business; the use of templates adds defaulted values into many of the fields that have been chosen by the business as necessary.

A business that builds and maintains escalators may have two templates, one for new builds and one for maintenance, which default the Quick Entry screen data as shown in Table 14-2.

Note that Projects maintains the duration of a project in the template, so if the template project has a duration (end date minus start date) of 30 days, when a start date is entered in the Quick Entry screen, the end date will default to 30 days after it.

Transaction Setup

The crux of the Projects application is in tracking transactions against the project and tasks that are set up. There is a great deal of categorization required in Oracle Projects setup to support the wide array of project types and costing and billing scenarios. This makes the module extremely flexible, because reporting, budgeting, and accounting can all be tailored to business needs. Financial movements on a project are grouped as shown in Figure 14-2.

The bottom layer of this structure—event classification and expenditure-type class—are seeded values; the rest are user configurable (see the expenditure-type setup in Figure 14-3).
### Table 14.2. Quick Entry Screen

<table>
<thead>
<tr>
<th>Quick Entry Field</th>
<th>New Build</th>
<th>Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Number</td>
<td>NB1</td>
<td>M1</td>
</tr>
<tr>
<td>Project Name</td>
<td>New Build</td>
<td>Maintenance</td>
</tr>
<tr>
<td>Project Start Date</td>
<td>Nov 2011</td>
<td>Nov 2011</td>
</tr>
<tr>
<td>Project Completion Date</td>
<td>Jul 2012</td>
<td>Nov 2016</td>
</tr>
<tr>
<td>Project Description</td>
<td>New Build</td>
<td>Maintenance</td>
</tr>
<tr>
<td>Project Status</td>
<td>Submitted</td>
<td>Submitted</td>
</tr>
<tr>
<td>Public Sector Indicator</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Organization</td>
<td>Birmingham Engineering</td>
<td>Camberley Maintenance Depot</td>
</tr>
<tr>
<td>Customer Name</td>
<td>London Transport</td>
<td>London Transport</td>
</tr>
<tr>
<td>Key Members (by project role type)</td>
<td>Project manager—S. Jones</td>
<td>Project manager—R. Smith</td>
</tr>
<tr>
<td></td>
<td>Site manager—R. Atkinson</td>
<td>Maintenance manager—A. Capp</td>
</tr>
<tr>
<td>Project Classifications (by class category)</td>
<td>Contract type—new build</td>
<td>Contract type—maintenance</td>
</tr>
<tr>
<td></td>
<td>Market sector—public</td>
<td>Market sector—public</td>
</tr>
<tr>
<td>Distribution Rule (by project type)</td>
<td>Fixed Price—cost/event</td>
<td>Time and Materials—work/work</td>
</tr>
</tbody>
</table>

**FIGURE 14-2. Transaction categorization**
Transactions that come from other Oracle modules, such as inventory movements or supplier invoices, have expenditure type classes assigned. They will also have the expenditure type on them because project, task, and expenditure type are the three keys required on entry of these transactions in the source modules. Taken together, the six elements in this structure offer enough granularity to satisfy a rich variety of reporting needs.

An example of how part of this structure might be used is that straight time and overtime can be posted to the same GL account using the expenditure category of labor to determine the account in the AutoAccounting feature. In the Projects module, reporting by expenditure type can still allow effective management of the overtime bill.

One other part of transaction setup is resource groups and resources. This type of categorization is used for budgeting and for summarizing data for the Project Status Inquiry screen. There are three layers to this structure: resource lists, resource groups, and resources; resource groups are optional. When you undertake a project, you may use labor, services, materials, or equipment. Such expenditures can be reported in many ways. For example, you might report an invoice for materials by supplier, expenditure type, or organization; these are resource types, and the individual values are the resources (for example the suppliers or organizations themselves). You can optionally further group the resources into groups by expenditure category, revenue category, or organization. You can attach either the resources or

FIGURE 14-3. Expenditure-type setup
the resource groups to a resource list. You attach resource lists to a project so you can use them to set the project budget and report actuals against budget in the Project Status Inquiry screen. You can use the same resource lists across similar projects to enable cross-project comparisons.

Transactions imported from other systems are allocated a transaction source. This source determines whether the transaction has already been accounted for in the GL. For example, you can import payroll transactions into Oracle Projects for reporting purposes but also import summary transactions into the General Ledger to be accounted. To avoid double accounting, set the transaction source to “pre-accounted” so that the transactions will not be picked up by AutoAccounting.

Another option on the transaction source is to import an expenditure organization that is different from the employee-owning organization. This means employees can be allocated to a project that is not in their organization and the cost can be tracked. The employee number is on the transaction record, so for reporting purposes you can still report on that employee’s activities.

**Costing**

In the course of a project, costs accumulate; they are of many different types, depending on the type of project. Most projects will have the costs of purchases (whether bricks, subcontracted labor, or consumables). These costs will be identified at the requisition stage as being against the project (down to task and expenditure type). The Oracle Purchasing and Self-Service Procurement modules have the requisitioners enter project data in much the same way they enter General Ledger data. In fact, the project and task data, user identity, and type of goods are usually enough to generate the full Accounting Flexfield.

The project’s commitment to buy is viewable in the Projects module, in the Project Status Inquiry screen. When the invoice is received, matched, and approved, the invoice lines are interfaced to Projects from Payables. At this point the commitment decreases by the amount of the invoice and the invoice appears in Projects as an expenditure item.

Many projects will use facilities such as computers and machine tools. The usage of these can be assigned to the project, task, and expenditure type. Usage logs can be prepared, signed off, and then keyed into Oracle Projects as expenditure items, as shown in Figure 14-4. Alternatively, they can be imported through the transaction-import interface. This is a standard Oracle open interface.

The same interface also supports the import of other types of costs—for example, labor costs. Very often these will be imported from a payroll or time-sheet system, which contains the project, task, and expenditure type that relates to each labor hour spent. These straight-time transactions are associated with employees who must be set up in Oracle’s Human Resources tables (although they do not need to be linked to the project as key members). The HR tables are accessible through screens in Oracle Projects Responsibilities if Oracle HR is not installed as a full install. Unlike the HR screens in Payables, those in the Projects Responsibilities give assignment-history functionality so that the employee’s job and location can be tracked (a useful feature for Projects reporting).

As the E-Business Suite is aimed at an increasingly globalized market, the capability to handle multiple currencies is important. Projects are no exception, and costs can be received in currencies other than the project currency (in which project reporting is done). There is a currency specified for the project (project currency), for the operating unit (functional currency), for the transaction (transaction currency), and for Self-Service Expenses (reimbursement currency).

Costs do not have to be just received or inputted to the system through the screen (shown in Figure 14-4). Oracle Projects can calculate indirect costs through the burdening concept. Burden costs are overhead estimates, as an alternative to allocating costs collected on indirect projects.
Burdening is sometimes known as cost plus processing. The burdened cost is the raw cost plus the burden cost. Raw costs, categorized by expenditure types, are grouped into cost bases.

For example, you can group expenditure types such as project management, consulting, and clerical into the office-labor cost base. Next, you define burden-cost codes to cover the types of overhead—for example, employee benefits and office space. Finally, you create a schedule to hold the multipliers for each of these cost codes—for example, 0.4 and 0.3, respectively. Say the labor cost base for a project amounts to 100; the system will calculate the burden cost as $(100 \times 0.4) + (100 \times 0.3) = 70$. Therefore, the burdened cost of the labor cost base will be 170.

You can account for burden costs separately, count them in with the raw costs, or not account for them at all but just use them as a management accounting tool.

Allocations
The burdening process estimates overhead costs and adds them to actual incurred costs. The allocations feature takes incurred costs that you decide are overheads (indirect) and allocates them to projects. Thus you can use the power of the Projects module to collect and report the details of costs on indirect projects, then allocate those costs to your capital and contract projects. You can also use allocations to charge specific fixed amounts or even General Ledger account balances. You can still use burdening even if you use allocations.
Take, for example, a telemarketing agency that has a four-floor office block. Each floor is dedicated to running marketing campaigns for different clients. The agency has set up a contract project for each client and a task under the project for each campaign. The rent for the office is posted to an account in the General Ledger. The agency defines a rule that allocates the rent paid on the office to the projects that use it. Cross charging has been enabled because the organization that rents the building is not the same as those that use it.

Another example is where the information-systems (IS) department runs as an indirect project. The IS costs are collated, including time-sheet information, usage of computer resources, and hardware and software costs. Only some of these costs are allocated—those that relate to the labor time of the support staff and the software-license costs, but not the hardware costs.

Allocations are rule based. You can create a number of rules, each of which performs a different set of allocations. Rules define from where the allocation is sourced, what its targets are, and how it is spread (see Figures 14-5 and 14-6). The source can be a project, just a task from a project—even restricted by resource—or a GL account.

Alternatively, a specified flat amount can be allocated among the target projects and tasks. The targets are specific projects or tasks (or whole projects excluding specified tasks). The allocation can be spread evenly across all projects and tasks specified or it can be targeted, say 80 percent to one project and 20 percent to another. The prorate method uses a resource as a basis for apportioning. This can be used to apportion the cost of a machine by the usage hours of each target project. If you want, you can set up an offset that creates a reversing expenditure.
(meaning a negative cost) on the designated project and task (which could be but does not have to be the same as the source). This can be useful to monitor the amounts allocated. (Note that it will decrease the total expenditure on the offset project.)

Once rules have been created for the allocations required, they can be built into Auto Allocation sets. A set is essentially a request set that can run a number of different project allocations, and GL mass allocations too. If you have many allocations in a set, you can have it run either in parallel or in a specific sequence as a step-down allocation set (aided by Oracle Workflow). When a set runs, it creates a draft release of allocation transactions for review. If you are not happy with the rule, there is an option to delete the draft, change the rule, and rerun. When you are happy with it, you run a release process that creates an expenditure batch with costs allocated to the target projects.

**Billing**

There are two functions that are only available to contract projects: accruing revenue and billing, for which the accounting flows are different. Accrued revenue passes from Projects straight to the General Ledger; the billing amount and details pass into Receivables on a draft invoice (which might be in the project currency or the project-customer currency). AutoAccounting provides the account-code combinations for both sides of the transaction. When the invoice has been processed in Receivables, the receivable amount is interfaced from there to the General Ledger.
Certain preconditions have to be met before revenue will be accrued on a project. Obviously it must be a contract project; it also needs to be in the correct status and have funding. It is possible to have a funding arrangement in place that imposes a hard limit, much like encumbrance does on purchasing, so no further revenue will accrue once the limit has been reached.

Revenue can be accrued against projects where billable work has been completed but not yet invoiced. Revenue accrual allows internal management reporting to show realistic figures for revenue and still honor the billing and invoicing schedule agreed upon with the project customer (see the following illustration).

<table>
<thead>
<tr>
<th>Unbilled Receivables</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) $150</td>
<td>a) $150</td>
</tr>
<tr>
<td>b) $150</td>
<td></td>
</tr>
</tbody>
</table>

Contract projects can accrue revenue using several different models. You define which one is to be used at project definition by way of the project type and distribution rule, as shown in Table 14-3.

As you can see from the table, there are two different ways of accruing revenue: by expenditure item or by event. These methods map to the different types of project billing.

<table>
<thead>
<tr>
<th>Distribution Rule</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost/Cost</td>
<td>Accrue revenue and bill using the ratio of actual cost to budgeted cost (percent spent).</td>
</tr>
<tr>
<td>Cost/Event</td>
<td>Accrue revenue using the ratio of actual cost to budgeted cost (percent spent); bill based on events.</td>
</tr>
<tr>
<td>Cost/Work</td>
<td>Accrue revenue using the ratio of actual cost to budgeted cost (percent spent); bill as work occurs.</td>
</tr>
<tr>
<td>Event/Event</td>
<td>Accrue revenue and bill based on events.</td>
</tr>
<tr>
<td>Event/Work</td>
<td>Accrue revenue based on events; bill as work occurs.</td>
</tr>
<tr>
<td>Work/Event</td>
<td>Accrue revenue as work occurs; bill based on events.</td>
</tr>
<tr>
<td>Work/Work</td>
<td>Accrue revenue and bill as work occurs.</td>
</tr>
</tbody>
</table>

**TABLE 14-3. Distribution Rules**
available. Expenditure items on a contract project can be billable, and the system calculates how much revenue to accrue based on different formulas, which depend on the distribution rule.

Time-and-materials projects accrue revenue for the expenditure items (labor and materials) purchased. The amounts charged are determined by schedules of bill rates. These can be specific to employees, set up on the jobs they hold, or associated with nonlabor resources like machine tools or computer equipment.

Fixed-price projects can be set up to accrue as they spend. On the cost/cost rule, the project accrues a sum related to the percentage of total burdened costs that have been spent so far, calculated as in the following illustration.

<table>
<thead>
<tr>
<th>Actual burdened cost to date = $5,000</th>
<th>Total budgeted burdened cost = $20,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total budgeted revenue = $100,000</td>
<td>Event revenue to date = $0</td>
</tr>
<tr>
<td>Revenue previously accrued = $15,000</td>
<td></td>
</tr>
</tbody>
</table>

The formula for calculating the increment of revenue to accrue would be:

\[
(\frac{5,000}{20,000} * (100,000 - 0)) - 15,000 = 10,000 \quad \text{(provided there is funding available if a hard limit is being used).}
\]

Percent-complete projects are another way of accruing revenue for a fixed price. You can enter percent complete against any task in the work-breakdown structure, but revenue accrues against top tasks or at the project level, whichever you have specified. You record the percent complete and the as-of date, and when you come to generate revenue, the system will calculate the percentage of the final budgeted figure based on the percent complete of each billable task.

Events can be used to do all or part of the billing for a project. Events can drive the revenue and billing for a milestone-driven project or can be used if the customer pays a performance bonus. They can be manually inputted and thus give control to the user over the revenue and billing amounts separately.

**Intercompany Projects**

Projects do not necessarily respect the organizational boundaries that divide a business. For example, you might have a contract with a global customer to do oil-rig refurbishment. You will resource the project from several regional offices but want to keep it as one project to manage the costs and reporting to the client. As you charge work from the regional organizations to the project owner (say, Engineering HQ), you can choose from two methods (if you want it done in Projects; otherwise it can be done in either the General Ledger or a third-party system).

The Eastern Region is in the same legal entity but a different operating unit from Engineering HQ. It does not want to invoice for the work it does, so it can use the borrowed-and-lent method. It sets up a transfer-pricing schedule for the work it does on the project. Eastern Region then charges its time and materials to the project, task, and expenditure type. The transactions will show Eastern Region as the expenditure organization. The Distribute Borrowed and Lent Amounts process determines the transfer-price amounts and the accounting entries. This process credits the Eastern Region costs and debits Engineering HQ’s costs. Finally, as the provider, Eastern Region runs the Interface Cross Charge Distributions to GL process that interfaces those entries to GL.
The Central America Region is in a different legal entity from Engineering HQ. Note that it still must be in the same business group and share the same GL and project-accounting calendars. It requires physical invoices to be generated, so it uses the intercompany-billing method. Engineering HQ will be set up to allow charges from other operating units and to receive internal billing. You set up the list of eligible provider operating units. The transactions start life being charged by the region to the receiver organization, as with the borrowed-and-lent method.

AutoAccounting enables you to specify cost-reclassification accounts during the cost-distribution process. So for example, costs on the job charged by Central America go to a Work in Process (WIP) account but are reclassified as a labor expense, because this work in process is not an asset on the region’s balance sheet. The Central America Region now runs the Generate Intercompany Invoices process. After their approval and release, the invoices are interfaced to Receivables (from where they can be printed).

The Tieback Invoices from Receivables process is used to create the corresponding supplier invoices in the Payables interface table for Engineering HQ, which can import them. After approval, these Payables invoices are interfaced to Projects, as they are coded to a project, task, expenditure type, and expenditure organization.

**Interproject Billing**

The principles of intercompany billing are also applicable to projects within the same organization. For example, a design house might want to manage individual design jobs as separate projects for cost control and reporting. However, their client might be a large automobile manufacturer that is unlikely to want to be billed for each one individually; therefore, the design house can either bill from one of the projects involved or set up a consolidation project. The project that bills the client is set up as the receiver, and the others as the providers. This is effectively a subcontracting relationship.

The providers generate Receivables invoices; then the Tieback Invoices from Receivables process creates payables invoices for the receivers. These payables invoices will be interfaced into the Oracle Projects module associated with the receiver project, task, and expenditure types. On the basis of these charges on the receiving project, external invoices can be generated to send to the customer that cover the activities of all the providing projects.

**AutoAccounting**

Oracle Projects is a projects subledger. It receives transactions from surrounding business processes that relate to the projects set up within it. These transactions might include the following:

- purchase requisitions for raw materials, consumables, or subcontracts;
- issues of the same from a warehouse;
- labor hours spent on project tasks;
- invoices received from suppliers for the supply of goods and services related to the project; and
- hours of use of a specialist machine—for example, to lay railway track.

Transactions that originate from outside Projects but within the E-Business Suite have their accounting set in the originating module. Purchase requisitions are accounted for by the account generator in purchasing. The account generator can use the project, task, expenditure type, item,
and requestor to determine the accounts. Inventory transactions pick up their accounting when the transaction is created.

Labor hours generally come from a third-party product; they can be preaccounted, with summary transactions sent to GL separately, or can be accounted for by AutoAccounting. Oracle Payables accounts for supplier invoices. Usage logs can be preaccounted or you can use AutoAccounting, the automatic account-generation engine used in Oracle Projects.

Batch processes do the bulk of the accounting work in Oracle Projects. The main areas are distributing costs, accruing revenue, and creating draft invoices. These processes use functions to determine which accounts should be debited and credited for a given individual cost or revenue line. The functions themselves are further subdivided into function transactions. For example, there is a function called “Labor Cost Account Function.” This is divided into function transactions, including

- private billable labor;
- all labor;
- private nonbillable labor;
- capital, all;
- public billable labor;
- contract, all;
- public nonbillable labor; and
- indirect, all.

Each of these function transactions relates to a single Accounting Flexfield combination. They are optional, depending on whether the business wants to distinguish between private- and public-sector work in the GL or report within Projects based on project classifications. If no distinctions are required, the “All Labor” transaction can be enabled, all the others disabled, and only that one mapped in AutoAccounting.

The structure used to determine the combination is rule based and user defined. It can use parameters (Rule A in Figure 14-7), SQL statements (Rule B in Figure 14-7), or constants (Rule C in Figure 14-7). You assign the rules to the function transactions; therefore, when Projects runs a process such as Distribute Labor Costs and it needs to find the labor-cost account for a labor-cost record, it uses one of the function transactions, which in turn uses rules to determine each segment value of the code combination.

In Figure 14-7, the business wants to determine the labor-cost account for all labor in the same way. It has a three-segment Accounting Flexfield (company, cost center, and account). Rule A derives the company segment from the expenditure organization on the labor-cost transaction. Rule B takes the value in a Descriptive Flexfield on the transaction and puts it in the cost-center segment. Rule C sets the account segment to a constant value of 5500.

This is undoubtedly a difficult structure to conceptualize; however, it is vital to do so because AutoAccounting is the engine that drives the accounting done by Projects. During the design phase of a Projects implementation, the challenge is to document the business requirements and the AutoAccounting solution so that business representatives will be confident in signing up to it.

By the time you perform integration or system testing, you need to decide the function transactions, rule assignments, rules, and segment pairings (associating rules with Accounting Flexfield segments). This way the test cycle will test the AutoAccounting functionality and you will have realistic postings in the General Ledger, which will provide a sense of security for the business representatives involved.
Project-Processing Cycle

The complexity of setup and the range of transactions that are inputted in Oracle Projects means the software has a great deal of number crunching to do to convert the raw transactions into costs and revenue. Therefore, it has been designed with a large number of batch-processing steps. For example, when an expenditure batch is inputted, it might include the number of hours of machine usage. The system records the raw data but the costs are not applied until the Distribute Usage Costs program is run to distribute usage costs; AutoAccounting determines the accounts required.

For example, the Distribute Usage and Miscellaneous Costs process takes the quantity on the transaction, applies the raw-cost rate from the nonlabor-resource setup, calculates the burden cost, and adds it to the raw cost to get the burdened cost. It then uses the usage-cost-account function to determine the appropriate cost account to add to the cost-transaction record.

The transaction sources for the costs being interfaced from other modules are seeded as preaccounted. However, if adjustments can be done in Projects (such as on supplier invoices), these will need to be accounted for by AutoAccounting; therefore, there is a Distribute Supplier Invoice Adjustment Costs process. Once the various types of cost are distributed and accounted for, they can be interfaced to the General Ledger.
Chapter 14: Oracle Projects

The revenue side is also controlled by concurrent processes. Cost transactions can also be a source of revenue, so there is a Generate Draft Revenue process, which calculates the revenue available on each cost line and revenue event and determines the revenue accounts.

When the costs have been distributed and the draft revenue generated, you run the Interface Revenue to GL, Generate Draft Invoice, and the Interface Invoices to Receivables processes. These do the accounting for unbilled receivables, receivables, and unearned revenue.

The following is a list of some of the processes that use AutoAccounting. This is just a subset that is available to be run—it might be that a particular implementation will not use processes such as expense reports or burdening. The list shows the way the module is organized.

- Distribute Labor Costs
- Interface Expense Reports to Payables
- Interface Labor Costs to GL
- Distribute Usage Costs
- Generate Draft Revenue
- Interface Usage and Miscellaneous Costs to GL
- Interface Revenue to GL
- Distribute Total Burdened Costs
- Distribute Expense Reports
- Interface Total Burdened Costs to GL

Another process is required to populate the Project Status Inquiry summary tables. This should be done regularly (possibly every night), because runs will take much longer and use more computing resources if the frequency is reduced, therefore potentially squeezing the rest of your nightly batch-processing window.

The other regular set of processes is the interface between Oracle Projects and the other Oracle EBS modules. There is an option to submit processes together in a streamlined process, much like a request set. There are two types of these—“PRC:Submit Interface Streamline Processes” and “PRC:Submit Project Streamline Processes”—within which there are many options.

**Reporting**

To get the most out of the effort of loading all your data into a relational database such as Oracle’s, think about reporting requirements early on in the implementation. You will gain the most from your information if it is in buckets large enough to give valid statistical comparisons. For example, if one of your lowest-level tasks in testing is called “client test” in some projects, “business test” in others, and “user test” in yet more, reporting on it will not reveal any trends. However, if you insist on one task name for all like projects (embodied in a template), the analysts or managers with responsibilities across several projects will soon see trends and exceptions.

**Project Status Inquiry (PSI) Screen**

One of the most powerful reporting utilities for everyday use is the Project Status Inquiry form. It is built to use a series of accumulation tables that are populated by a batch process called Update Project Summary Amounts. This means that although information on projects is not
reflected in this screen in real time, the screen does not need to do online calculations and therefore is quick to use.

The process populates four time dimensions—inception to date, year to date, prior period, and period to date. In these dimensions, budget and actual figures are shown (and commitments), for both costs and revenue (see Figure 14-8). All of this is held within the structure of project, task, and resource. The power of the PSI screen is that it provides a drill-down feature from the project, through the task structure, to resources, even to the revenue and cost transactions themselves. The Export to Excel function makes this a good tool for analysis and presentation. You can use the folder technology on the PSI screen to alter the available data so you can focus the information for particular audiences.

There is a series of database views that display data from the underlying summary tables. The top layer of views is user customizable in the Project Status Inquiry setup screen (see Figure 14-9) by the use of SQL statements such as the condition-processing statement DECODE. Clearly this kind of customization should be constrained as far as possible to the implementation phase of a project; during live running, you need to provide consistency of reporting over time to make good use of comparative information.

In the default folders there are 27 columns for the project and task levels, and 29 columns for the resource. These include the database columns, such as actual and budget cost and actual and budget revenue, and calculated fields such as margin and estimate to complete (in financial and hours of labor terms).

Oracle Discoverer for Projects
Oracle Discoverer provides some pre-built business areas as a reporting solution for Oracle Projects, called Oracle Project Discoverer, which consists on:

- **Staffing**  The staffing business area provides general staffing information about resources, projects, and organizations. From a supply perspective, you can identify information regarding resource and organization capacity, availability, overcommitment, and overall schedule through these folders. From a demand perspective, you can identify information regarding project and organization requirements.

- **Financials**  The financials business area provides information about the project pipeline.

- **Competence**  The competence business area provides information about the competence of the resources.

- **Utilization**  The utilization business area brings a Business Intelligence model composed of dimension and data folders. Some dimension folders provided by Oracle Projects
Discoverer are Person, Work Type, Job, Operating Unit, Utilization Organization, Supervisor Hierarchy, Time Periods, Number of Trend Periods, and Time Trend. Data folders contain the resource and organization utilization information. The following data folders are provided: Manager-Resource Utilization, Manager-Resource-Worktype Utilization, Organization Utilization, Organization-Job Utilization, Organization-Resource Utilization, Organization-Worktype Utilization, and Organization-Resource-Worktype Utilization.

These reports, or also called workbooks within Oracle Discoverer, use data from the End User Layer, which is essentially a series of database views that provide simplification of the transaction-processing tables for reporting. Because Projects tables have been included in the End User Layer, you can add your own custom reports to the workbooks without the need for a developer.

**NOTE**

The latest Oracle Discoverer release is 11g; some parameters in Oracle Projects Discoverer are only available when using Discoverer 4i, so you will have to validate the reports, and in some cases a customization will be needed.
Integration with Other Modules

Oracle Projects can be implemented as a stand-alone module, as it has sufficient GL setup screens and HR setup screens to populate the required data in those shared tables. However, its integration with other Oracle E-Business Suite modules is very strong (see Figure 14-10). It is this integration that provides businesses with the biggest opportunity to increase efficiency. For example, in a full implementation of all modules, the time used to create customers is reused on every separate transaction with them, whether you are checking their credit status in Receivables, receiving orders for spare parts through Order Management, or receiving a contract project.

Oracle Projects casts the additional dimension of project on the rest of the E-Business Suite. As noted previously, projects are different types of entities from those the standard financial-reporting tool—the General Ledger—is set up to handle. Projects are of limited duration; they can be divided into tasks that are far more granular than is usual in Ledger reporting; and inception-to-date reporting is very important for projects.

Integration takes two forms: some data are viewable from one module to another; some data are passed in batches between modules. All modules in the suite refer to Oracle Projects to validate project information. Many use project and task data as input to Automatic Account Generation. See Figure 14-10 for the complete picture of module integration.

Reference Data

General Ledger referencing is implemented in Projects as it is elsewhere in the E-Business Suite. Accounting Flexfield combinations are accessible and the standard APIs are used by the interfacing processes to comply with cross-validation rules. In the Projects super-user Responsibility you can see the GL setup screens.
Human Resources information is referenced extensively in Projects, and you can see some of the screens in the Projects super-user Responsibility, although if you have HR installed you will not be able to use those screens, as more complete versions are available in the HR Responsibilities. The organizations, organization hierarchy, and employee setup are all shared between Projects and HR. It is particularly in this area that a coordinated approach in the implementation phase will have benefits.

In a business organized on project lines, it is important for the entry of data against projects to start right at the beginning of the transaction flow. Employees working on a project will book their time to it. They might use Oracle Self-Service Time to do that, in which case when they do their time sheet online they will specify the project details, which are validated on entry. Self-Service Time is a Web application that uses the same technology stack as Self-Service Procurement and Expenses. Users can book their time to projects either online or on an Excel spreadsheet that is subsequently uploaded.

Workflow is used to approve time sheets (or it can be set to automatic by a profile option, if preferred). The core functionality of the Workflow application has a long pedigree internally in Oracle and delivers significant efficiency benefits. Because it can be implemented across the Internet, there is no client software to maintain and it can be accessible from any Internet-connected browser, without the plug-in required when using the screens of the E-Business Suite professional interface.

While using Purchasing and Self-Service Procurement, users will make project-related purchases, such as materials or subcontracts. When the invoices are received for the materials or subcontracts, the users can match the invoice to the purchase order and the project-related information will be carried through (without the option to overwrite it in Payables). They will issue materials from an inventory to a project and might have manufacturing costs that are associated with a project. The standard key on these transactions is the project number, task number, expenditure type, and expenditure organization. This is because they all use the same entry point to Projects—the transaction interface.

The Transaction Interface

The same interface is used in the E-Business Suite to collate costs from other modules and external third-party systems. It distinguishes among the various sources by the transaction source. New transaction sources can be set up, but when you do that you will see that those used by the intermodule processes are seeded and unchangeable.

The main features of the transaction import interface are as follows:

- loads transactions to project, task, and expenditure type;
- can load transactions from different expenditure organizations;
- can reference, on transactions, employees set up in HR (the interface validates them);
- can load multicurrency costs; currency can be the same as the project or the functional currency of the expenditure operating unit;
- populates a single open interface table: PA_TRANSACTION_INTERFACE_ALL;
- loads by batches; if one record in a batch errors, the whole batch errors;
- can load transactions whether accounted or not (determined by transaction source); those accounted must have code combinations and will not be processed by AutoAccounting;
- can load transactions whether costed or not (determined by transaction source); costed transactions must have a cost, whereas uncOSTED ones have only quantities and will have costs calculated by the appropriate distribution process;
can load costs whether burdened or not; a cost can be already burdened or a separate transaction can hold the burden cost to be added to the raw cost; and

has a Review Transactions window in which errored transactions can be corrected for reprocessing (note that errors should be corrected in the source system wherever possible).

See Figure 14-11 for the whole process flow.
Integration with Oracle Workflow

There are two workflow processes out of the box for Oracle Projects. One is to control the change of project statuses (see Figure 14-12), the other the baselining of budgets. When a user creates a project and submits it for approval, Oracle Workflow will route the approval to the user’s supervisor (as defined on the employee record). If approved by the supervisor, Workflow changes the status. A similar process works for changing a budget from working to baselined.

Interfaces from Oracle Projects

Oracle Projects interfaces master data to the full E-Business Suite for validation. It passes transaction data to the General Ledger, Receivables, Assets, and Payables modules. To the General Ledger, Projects passes costs and revenue (including unbilled and unearned). There are three parts to the process: interface, import to GL, and tieback. In the interface process, Projects collects all the transactions that have been cost, revenue, or cross-charge distributed that are eligible for interfacing and loads them into the GL_INTERFACE table.

The GL date of the transaction is the end date of the earliest GL open or future period after the PA date. The PA date is the end date of the project-accounting period in which costs, revenue, and invoices are accounted for. AutoAccounting determines the relevant accounts.

Once transactions are in the interface table, GL import is run and creates GL journals, with a source of Projects and journal-entry categories of

- Labor cost,
- Usage cost,
- Total burdened cost,
- Borrowed and lent,
Oracle E-Business Suite Financials Handbook

- Provider cost reclassification, and
- Revenue.

The tieback process is then used to print reconciliation reports to track the interfaced records. There are audit reports that can also help with this task.

The interface to Receivables works similarly in that there are three stages: the Generate Draft Invoices process, which populates the RA_INTERFACE_LINES_ALL table; the Receivables autoinvoice run; and a tieback process. All three can be run using a streamline process. Oracle Projects uses seeded values for the batch source, transaction types, and line-ordering and line-grouping rules; these should not be changed in any way. After the autoinvoice program runs, the tieback process reports back on which records were successfully interfaced and which not, and also creates the lines in the Payables interface tables for interproject and intercompany charges.

The interface to Oracle Assets takes the cost lines created in Oracle Projects against an asset and sends them to the assets module when the asset being constructed goes into service. All the costs associated with the construction-in-process item need to be interfaced to the General Ledger from Oracle Projects; then the date placed in service can be set in the Capital Projects screen (in the Projects module).

The minimum information for the asset is to have a corporate asset book associated with it. You can either assign the category, location, depreciation-expense account, and asset number in the Capital Projects screen or, after the next step—interface to Oracle Assets—in the Oracle Assets Mass Additions screen. The process credits the construction-in-process account (to which the asset-related costs had been posted) and debits the asset-cost account. Projects prints reports on the interface to Assets; however, after the Post Mass Additions interface has run, there is no tieback report. If further expenses attached to the asset come in after the transfer, you can post them to Oracle Assets separately and merge them there with the original asset.

As an alternative to using Self-Service Expenses, you can input expense reports into Oracle Projects. To track and pay these expense reports, Projects interfaces them to the Payables Invoice Import open interface. The accounting transactions are then transferred from Payables to the General Ledger.

**Activity Management Gateway and Project Connect**

The Activity Management Gateway (AMG) is a set of APIs that facilitate loading Oracle Projects with data from third-party systems. It is a separate product and is licensed separately. Unlike open interfaces, it does not have a separate set of tables; instead it is a series of package procedures a developer can call from a custom program, which provide the validation that is done in other circumstances by an open interface (like the Receivables Customer interface).

The developer still needs to do the work to create a custom wrapper for the APIs, but within it just passes the required parameters to the AMG procedure and loads the data into the application tables. The AMG also has APIs for extracting data for interfacing out to third-party systems. Thus percent complete and earned value can be sent back to the originating system.

You can create projects with or without agreements, or update or delete them (if there are no transactions against them). You can create, update, or delete tasks and budgets. There is a Control Actions screen in Oracle Projects that gives you control over which actions can be undertaken in the module on records imported from third-party systems; for example, you can prevent users from deleting tasks that have been added using the AMG.

The AMG can be used as a legacy-data conversion utility or in live running. For data conversion it is a useful way of quickly uploading a very large number of projects and tasks. It has been used in the field to load hundreds of thousands of tasks across thousands of projects.
Chapter 14: Oracle Projects

You should remember that developer skills are still required to build the AMG APIs into custom programs. The skills required are PL/SQL and an understanding of the EBS environment.

Project Connect links Oracle Projects with Microsoft Project via the AMG APIs. This is often an enticing possibility, marrying the graphical representation of the popular project-management software with the ERP connectedness of the E-Business Suite. However, it must be approached carefully, because it requires disciplines in a PC-based environment that might not be currently prevalent. From Oracle Projects you can send a standing data catalogue including class categories and class codes, which the Microsoft product will then use during project setup. During the life of a project, tracking information can be sent from Microsoft Project to Oracle Projects, and the two can be periodically synchronized.

Client Extensions

Client extensions are PL/SQL packages that Oracle provides in the expectation that users will customize them. They are therefore supportable by Oracle Worldwide Support (unlike customizations). Some are specified alternatives selectable on screens; others are called by concurrent processes. There are 29 extensible PL/SQL packages, some of which comprise several extensible procedures, such as the allocations package. In that one you can specify how source, target, and offset projects and tasks are selected. Source, target, and offset are in separate procedures.

The procedures have sample code in them so you have a template from which to work. You will need to follow a normal system-development methodology to control the production of client extensions; analyzing and documenting the requirement, solution, installation, and use of the software created. It is recommended that you keep the modified code in a code-control system so changes can be version controlled. The packages will be replaced during upgrades, so you will need to go through and reapply changes you have made, using the code you have in your code-control repository. Although the extensions are already embedded in procedures and therefore have some of the structure preordained (such as the commit to the database), you will still need competent PL/SQL developers who understand the Projects application’s table structure.

Data Conversion

Legacy projects and tasks can be converted either manually or through the Activity Management Gateway APIs. One common problem with data converted from legacy applications is the need for a substantial data-cleanup exercise. Because much time is spent cleansing each record, the incremental time required to type it in is not very significant until the number reaches the thousands. However, it should be remembered that data conversion is not a one-time affair. It should be done in test and live. The more test cycles you can process with real data, which cover all types of projects from all the different parts of the business, the smoother and surer will be your live conversion.

During the data-cleanup exercise, project classifications can easily proliferate, as each project has different characteristics from the last. Because the project owners see each one as unique, it is difficult for them to see the corporate view and the operational one simultaneously. Therefore, a positive commitment to a standard classification set early in the implementation by a senior manager will encourage a classification of projects that will go on to give meaningful management information.

To create cost conversions is not unduly difficult for a professional Oracle EBS developer. However, any revenue-events conversion would need to be directly to applications’ base tables (PA_EVENTS), so is not supported by Oracle. The custom cost-conversion program will need to
populate the PA_TRANSACTION_INTERFACE table. Once your custom program has done this successfully, you will be able to run the PA: Cost Transaction Import for the appropriate transaction source. This runs by operating unit, so you will need to run it once per operating unit. One quirk of this program is that it stops processing a batch once it has found an error in that batch. The standard work-around is to make each line into a batch.

The revenue- and billing-events conversion can be a relatively simple program that will insert into the PA_EVENTS table. When you design these programs, you will need to decide whether the records will be preaccounted. If they are preaccounted, enter code-combination identifiers in the relevant column and set up the transaction source as preaccounted. This way the records will not be interfaced to the General Ledger. Thus, if they are already included in a legacy GL conversion, they will not be double counted. Conversely, if you want them to go through the normal link to GL through AutoAccounting, leave the “Transaction Source Pre-Accounted” flag null (meaning no) and the code-combination identifier null, and the first run of the transfer to GL will pick up the transactions and process them.

**Conclusion**

Oracle Projects is at the vanguard of many of Oracle’s visionary ideas. It demonstrates the big themes of that vision. The cost-collection mechanisms are increasingly self-service based; time-sheet information, employee expenses, and procurement all benefit from central control but enable distributed entry by the people who have the information required. Workflow and Account Generator work mostly outside Projects to ensure approval processes for the incoming transactions but are also used in project-status and budget management. Globalization is enabled by the strong intercompany and multicurrency features. The core accounting logic implemented in AutoAccounting makes flexibility the keyword and gives control to the business over the granularity of the information going to the General Ledger.

The variety of tools, from the Project Status Inquiry form to the Oracle Discoverer workbooks, illustrate the sharpening focus of the E-Business Suite on drawing data out of the system.

All these features work toward greater precision and visibility of transactions in Projects. They will attract a wide range of industries that might not have traditionally been project oriented, but can gain considerable control over costs and examine their margin in the project dimension.