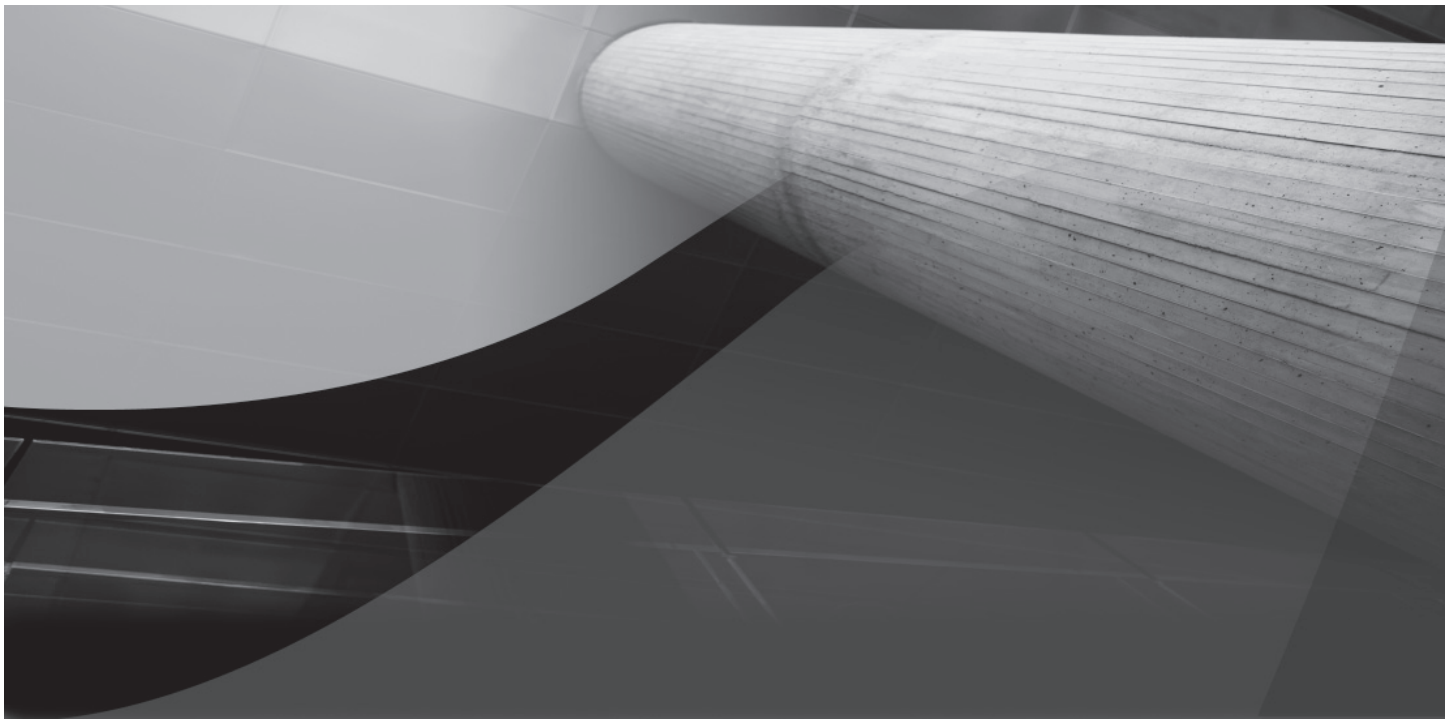


CHAPTER 9

Oracle Identity Manager

his chapter covers Oracle Identity Manager (OIM), which is central to Oracle's identity management strategy. OIM provides a platform for designing provisioning processes for user and access information to solve the challenge of getting the right accounts and privileges automatically set up for users across all applications they need to access. Configuration of such provisioning automation can be done in many ways; we'll show you some examples and best practice implementations of user provisioning. This chapter does not intend to cover the full set of OIM features and functionalities, and instead it highlights those that are used most frequently in solving provisioning problems.

The User Provisioning Challenge

When a new employee joins a company, she needs an office, a computer, office supplies, an e-mail address, access to business applications, and so forth. This process goes by many names—such as on-boarding or hire-to-retire. User provisioning is a subprocess initiated by the on-boarding or hire-to-retire process that deals specifically with giving users access to resources.

NOTE

"Resource" is a general term that can represent anything from a physical asset (such as computers, phones, offices, cubicles, printers, and so on) to logical assets (such as e-mail, payroll system, expense accounts, and so on). In the context of logical user provisioning, resources typically represent applications, databases, and other systems where accounts and privileges are set up for each user.

User provisioning has become a critical problem for most enterprises looking to lower their administrative burdens of account management while also trying to reduce risk by centralizing the control for granting access to important applications. Instead, with a user provisioning solution, new account creation tasks can execute in a consistent manner, whereby certain approvals and verifications are mandated before access is provided to new users.

The other critical user provisioning challenge is a technical one—system integration. A typical enterprise has a wide-ranging set of applications built on different technologies, standards, and semantics and therefore centralizing the account creation process is often an integration nightmare.

With these three drivers in mind—simpler administration, reduced risk, and easier integration—OIM was added into the Oracle Identity Management product suite with the acquisition of Thor Technologies, a smaller and best-of-breed user provisioning product. Since the acquisition, major development and improvements have been made on the product, but the basic framework and approach of user provisioning is still meant to be one that aligns with the three drivers.

Oracle Identity Manager Overview

OIM is a fundamental building block for an overall identity management solution. Access management, role management, directory services, and entitlement management all depend on having a working user provisioning solution that ensures the right identity data exists in the right location for other solutions to use. And with so many different types of policies, processes, and

integrations involved in a typical provisioning problem, the provisioning technology needs to support a high level of flexibility and customization. However, with added flexibility comes complexity, so OIM tries to achieve a balance between supporting customization of provisioning without making the implementation process too difficult.

The OIM product framework is architected in a way that allows the developer to choose the level of complexity to work with. Usually, a higher need for customization introduces higher levels of sophistication in configuration. For example, OIM provides many out-of-the-box standard integration solutions to connect into (in the form of packaged connectors and adapters) that provide basic solutions for OIM to provision into a particular system (such as Active Directory). However, additional requirements, such as approval workflows or custom attributes around provisioning, require that the developer customize the baseline connector to support those requirements. You will learn how to implement many of these requirements using OIM in this chapter.

Overall, OIM remains a sophisticated piece of technology that needs to be well understood before implementation. A good place to start learning about OIM is with some key concepts inside the OIM policy framework for managing user provisioning.

User

As described throughout this book, security entails understanding *who gets access to what* in any context. In OIM, a *user* represents that "who" in context of enterprise user provisioning. An OIM user is application-agnostic and, as such, can be provisioned to accommodate different applications using application-centric representations and data models. An OIM user defines a specific default data model with certain standard identity attributes, such as First Name, Last Name, Employee Type, Title, Organization, and so on, that can be extended as needed. The data model defines the fundamental enterprise-level identity data that drives the user's accounts and privileges in each resource.

User Group

In many applications, users are grouped together based on common functions, organization, job level, and so forth. OIM provides the *user group* object as a mechanism to support organizing users into simple compartments according to certain rules and policies.

A user can be associated to a group in two ways: via direct membership assignments or rule-driven memberships. Direct assignments are the intuitive mechanism with which most people are familiar. Membership is simple, straightforward, intuitive, and easy to understand and validate. Direct assignments are performed in a discretionary manner by another privileged user (such as administrators, managers, and so on), and the memberships are maintained in a static way (memberships are also revoked in a discretionary way). Direct assignment is therefore not a popular approach for certain applications and groupings.

Instead of static group memberships, you can use the notion of *membership rules* to manage group memberships in a more automated manner. Membership rules are simple conditional statements that are evaluated against each user to determine whether or not the user belongs to a group. Figure 9-1 shows a membership rule, "location == San Francisco." This is an example of automating group memberships based a "location" attribute value.

This rule defines its members as users who have a job title of DBA and who work in the San Francisco area. User groups using membership rules are more dynamic in nature and provide significant flexibility for managing *who* belongs to *which* groups and therefore should be granted *what* resources. This mapping is performed inside an access policy (discussed a bit later).

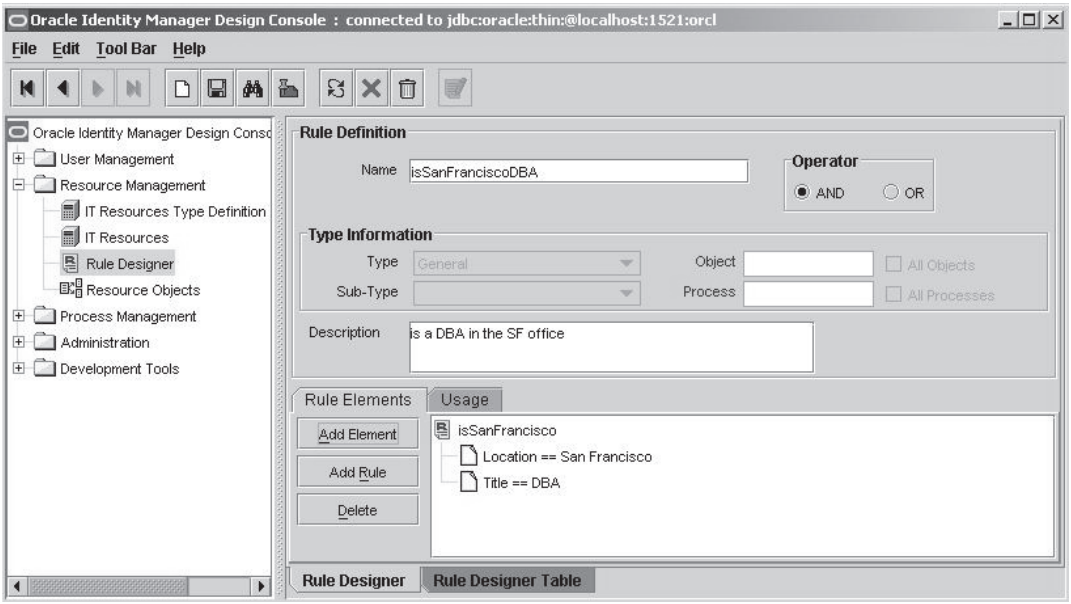
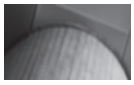


FIGURE 9-1 *Configuring membership rules for user groups*



NOTE
User attributes change from time to time and likewise change their group memberships. This removes the task that administrators generally have to perform when making static group assignments.

OIM also offers the organization object to also group users. However, the two objects are meant to organize users with different purposes that are complementary. Typically, user groups partition users based on user attributes that are cross-functional and could exist across the enterprise in any organization or department.

Organization

An OIM *organization* is meant to represent a business function or regional department, such as Sales, Product Development, North America Business Unit, and so on. OIM organization objects can be nested and therefore represent real-world organizational hierarchies.

There are three types of OIM organizations: company, department, and branch. Here is how each type maps common real-world organizational models:

- *Company* objects represent autonomous business units that own multiple business functions typical in any firm (such as Sales, Marketing, Finance, IT, and so on).
- *Departments* exist underneath company objects to represent business functions (such as Sales, Finance, and so on).
- *Branches* exist underneath departments to provide additional groupings of users, typically by geography.

It is not always necessary to divide and model your organizations exactly the way your firm or enterprise is organized, since you may not always need that level of detail. Also, keep in mind that a real-world organizational model can change frequently, so you may not always want to align fully to those models if your provisioning and access policies do not depend on the user's organizational associations.

TIP

An organization is different from a user group because a user can have at most one organization, but it can have multiple user group associations at the same time. So if you want to model a matrix organization in which a user belongs to more than one functional department, you can use a user group object to model those relationships instead of the organization objects.

Access Policy

An *access policy* is a way in OIM to map who should have access to what resource. The overall mapping from the user to the resource can be made up of mappings from the user to user groups and from user groups to resources. Figure 9-2 shows an instance of an access policy that can be automated to provision end users to the appropriate resources based on the rules and mappings that are represented by the arrows from each object.

In addition to controlling the resource, you can also control each user's privileges within each resource by associating application-level privileges to user groups in the access policy. For example, suppose two user groups, "Data Analyst" and "Data Administrator," should both be provisioned to access the same database application but with different database roles (such as analyst and DBA). You can set that mapping of user group to database roles inside an access policy.

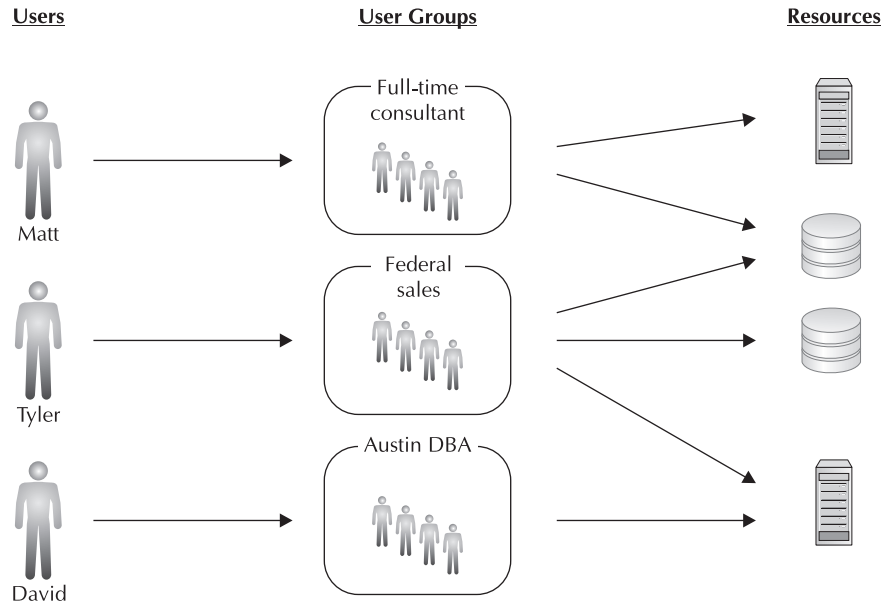


FIGURE 9-2 *Access policies define the mappings for users to groups and groups to resources.*

Resource Object

A *resource object* is an OIM object representing a logical resource for which users need to have accounts created. For instance, you can have OIM resource objects called "e-mail Server" and "Customer Database." A resource object can represent almost anything, from applications, databases, and operating systems, to physical assets and any other entity relevant to provisioning.

A resource object is used to track which users are provisioned to what logical assets. It can report on the current list of users who are provisioned to the E-mail Server resource in our example. Resource objects are also used to design approval workflows and policies around those workflows that are application-centric. So, for example, if a specific person is assigned to approve all new accounts to the e-mail Server system, you can use the resource object to set that condition in your workflow rule.

OIM resource objects do not represent the physical resources themselves and therefore do not contain physical details (such as IP addresses, server hostnames, and so on). For physical server representations and details, OIM provides the concept called IT resources.

IT Resource

An *IT resource* is a physical representation of a logical resource object. It holds all the physical details of the resource for which a new user is provisioned. If, for example, you have a resource object called Customer Database, you need to also define one or more corresponding IT resource objects that represent the physical characteristics of the resource (such as server hostnames, IP addresses, physical locations, and so on). This information is used by the OIM integration engine when it needs to communicate with those servers to complete a provisioning-related task.

The specific set of attributes of an IT resource is highly dependent on the type of system on which the account is being created (relational database IT Resources expect schema names and passwords; LDAP servers IT Resources expect names places and directory information tree details). OIM allows you to define an IT resource type that acts as a template to define a specific data model for certain types of IT resources.

User Provisioning Processes

A user provisioning process looks similar to any other business process. It represents a logical flow of events that deal with creating accounts within enterprise resources to make a new user productive.

Every provisioning process uses some fundamental building blocks, and the following sections provide different levels of sophistication in user provisioning. Your choice of sophistication level should, obviously, depend on the requirement and sensitivity of the particular resource. The level of complexity of a provisioning process is typically related to the level of risk associated with the resource being provisioned for access. For systems or databases holding critically sensitive data, provisioning should enforce a stronger verification process, such as requiring certain user attributes (such as job code or seniority) and management approvals before the user is granted access to an account in that critical system. Traditionally, these advanced provisioning processes were executed manually, but with OIM's process integration capabilities many of these provisioning enforcement tasks can be automated. The following sections provide some configuration solutions for the process-related challenges of user provisioning.

Discretionary Account Provisioning

Discretionary account provisioning is a style of provisioning by which an existing OIM administrator or privileged user can provision a user to an application in a discretionary manner. Inherently, a discretionary method is less consistent and leaves it up to the administrator to know what to do, rather than using a codifying a policy in the provisioning process. By default, this style of provisioning is automatically set up when an OIM is set up with an application using a packaged connector. And typically enterprises use this as a baseline to start designing and implementing their automation rules to make the process less discretionary. To provision a resource to a user in this manner, you'd use the Resource Profile For Users section in the OIM administrative console, shown in Figure 9-3, and click the Provision New Resource button to access the Provision New Resource wizard.

Typically, this style of discretionary provisioning is useful for enterprises that are looking to take the first step from manual provisioning processes to a basic level of automation and centralization. Also, if the enterprise lacks formal governance rules and policies around access

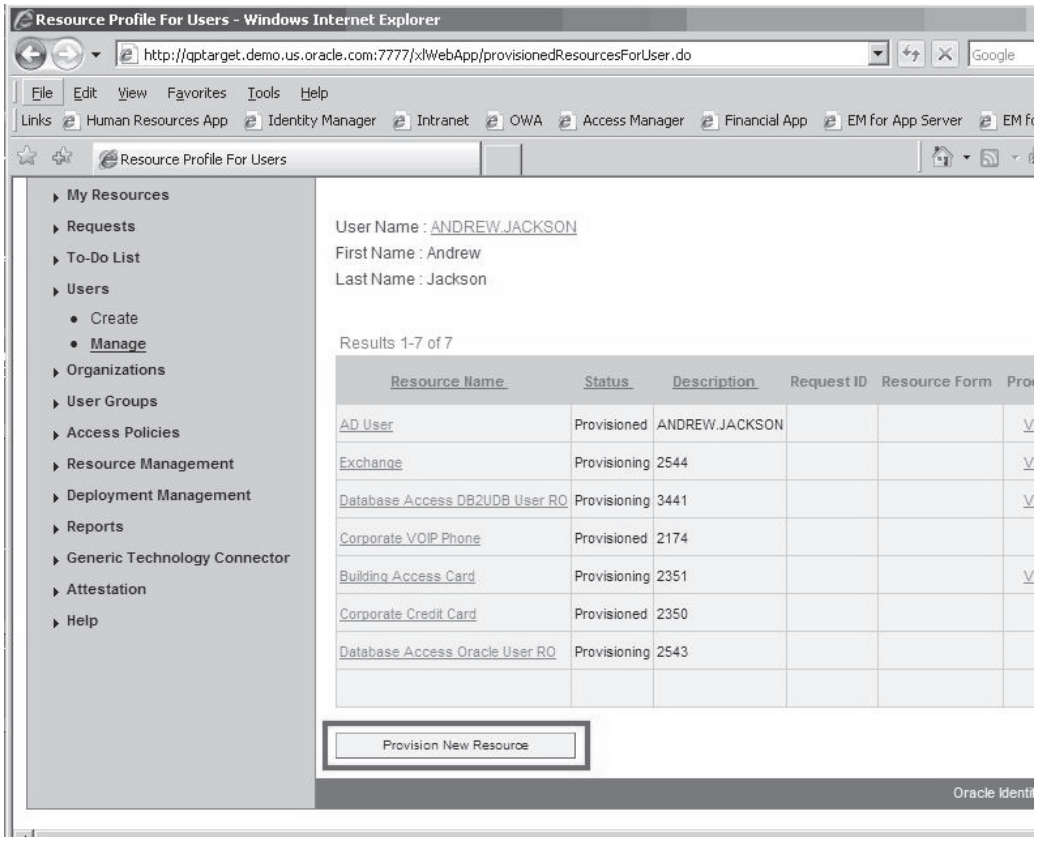


FIGURE 9-3 *Discretionary provisioning of new resources*

to systems and information, handling provisioning requests in a request-based manner might be the inevitable first step. However, if OIM has been put in place, you can accelerate your path to better provisioning automation by leveraging a lot of the built-in features of OIM, such as allowing users to make new requests through OIM and performing basic maintenance tasks such as password resets.

Self-Service Provisioning

The discretionary account provisioning requires an administrator or a privileged user to initiate the provisioning process. In other words, users will still need to make a phone call or send an e-mail to the administrator to request a new account in an application. However, OIM can be easily configured so that users can communicate entirely through the OIM framework when requesting access to new resources. To enable this, you need to set the Self-Service Allowed flag in the resource object for which you want to allow this. Figure 9-4 shows the option in the configuration screen.

Once this configuration has been set for a resource, that resource appears in the list of choices in the Request New Resource section in the OIM administrative console, as shown in Figure 9-5.

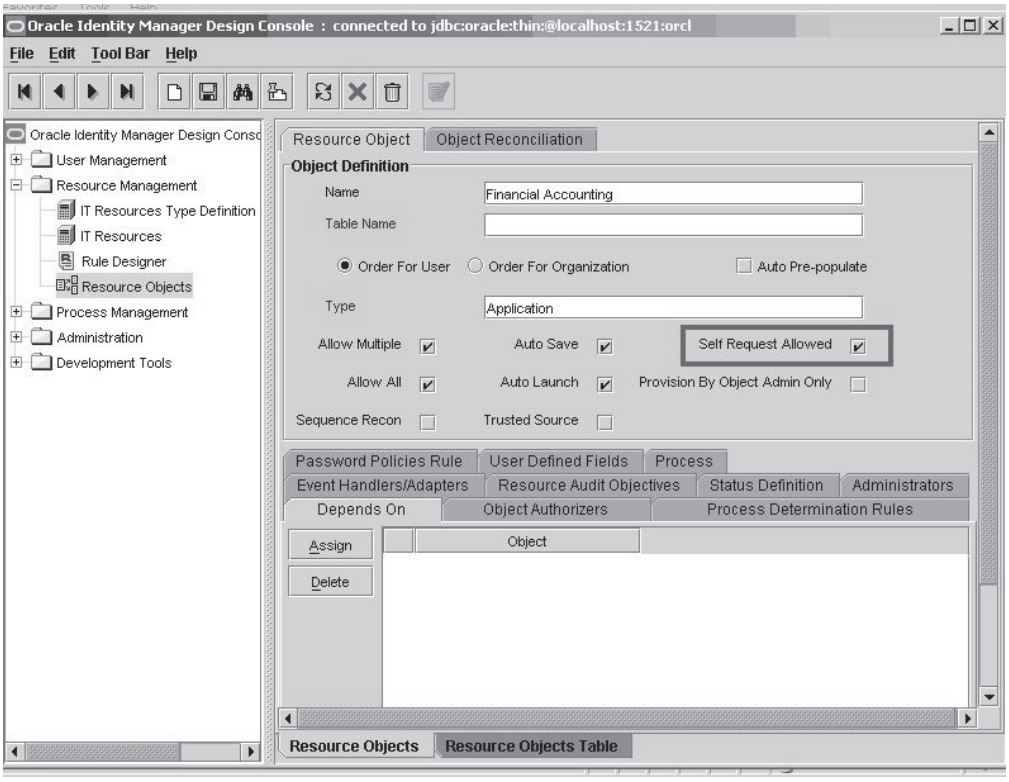


FIGURE 9-4 *Configuring self-service requests on resource objects*

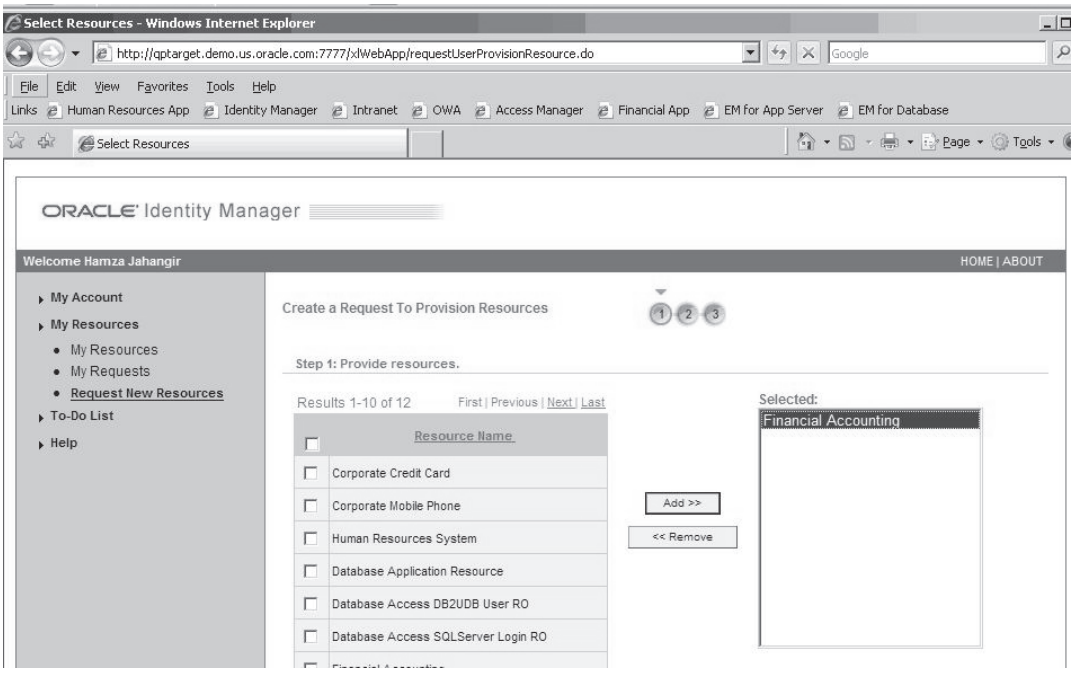


FIGURE 9-5 *Making requests for new self-service–enabled resources*

Over the past few years, self-service user provisioning has been a popular solution especially when delivering simple capabilities such as resetting passwords and requesting accounts in new systems and applications. It can greatly reduce the burden on administrators for performing highly repetitive tasks of manually inputting data from paper forms submitted by an end user. However, enabling the self-service capabilities on resources usually leads to some manual oversight, typically enforced through approval workflows that allow administrators to verify and sign-off on requests from end users. Without such approvals, the resource might as well be a fully public resource.

Workflow-based Provisioning

A workflow-based provisioning process gathers the required approvals from the designated approvers before granting a user access to an application or another resource. For example, the Finance application might require that every new account request be approved by the CFO to maintain tight control of who gets to see sensitive financial information.

To set up approval workflows, you can use the graphical workflow designer in the OIM administrator console, which you can navigate to from the following tabs: Resource Management | Manage | *Resource Name* | Resource Workflows | Create New Workflow

To continue the example from Figure 9-5, we'll create an approval workflow on the "Financial Accounting" resource object that requires two approvals: one from the user's manager and one from the application administrator. The following steps create the workflow:

1. Create a new approval workflow with a descriptive name.
2. Right-click the Workflow Designer and create a new task.

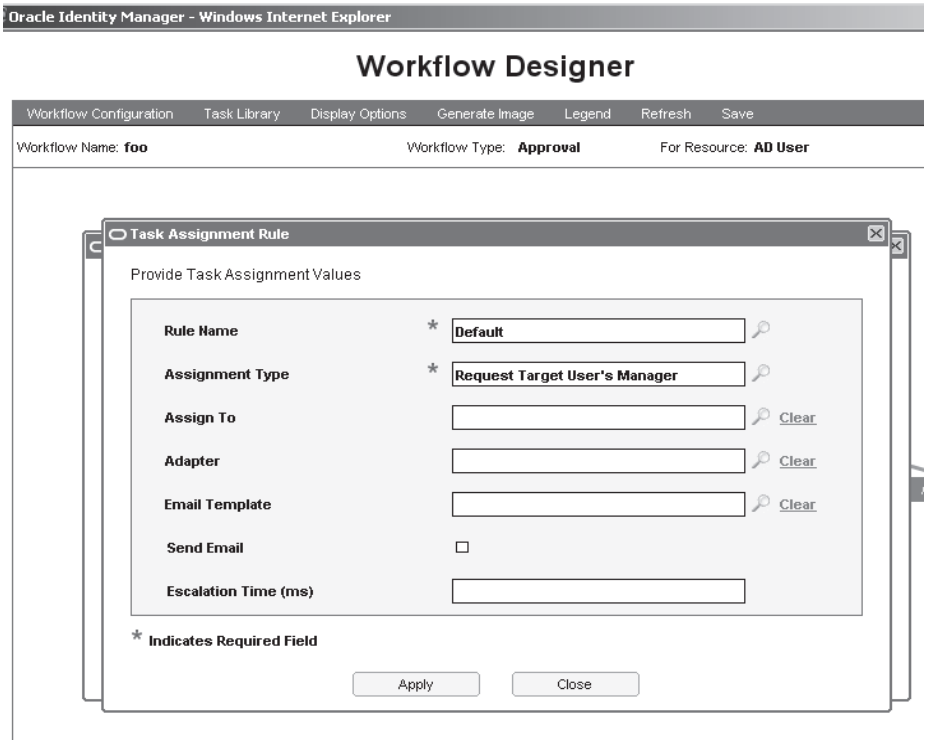


FIGURE 9-6 *Configuring an OIM workflow assignment rule*

3. Double-click the newly created task and go to the Assignment tabs.
4. Edit the Default rule and select the Assignment Type, as shown in Figure 9-6.
5. Select the Request Target User's Manager type, which is configured to route approval through the requesting end user's manager.
6. Once both the tasks are set up and configured appropriately, build the process sequence by right-clicking the Start icon and selecting Add Non-Conditional Task. Then drag the arrow to your first task (Manager Approval).
7. Right-click the Approve box of your first task, select Add Response Generated Task, and drag the arrow to the second task (App Admin Approval) to finish out the workflow. Figure 9-7 (on the next page) illustrates the completed view of this.

Access Policy–driven Provisioning

Recall the two keys questions that drive user provisioning efforts:

- Who *has* access to what resources?
- Who *should have* access to what resources?

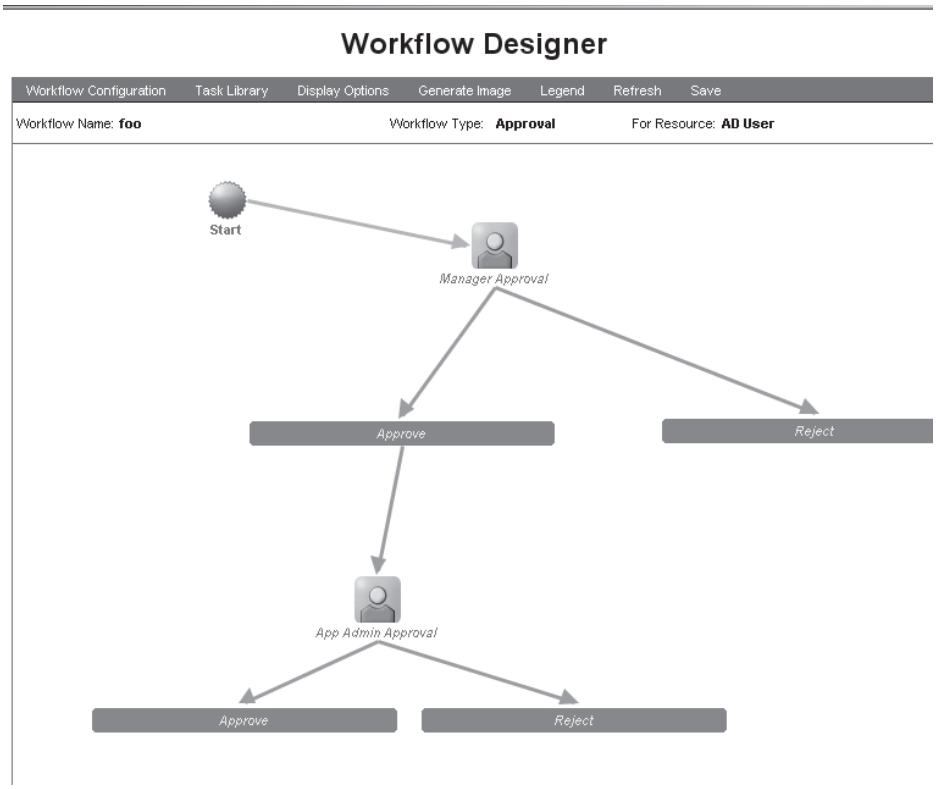


FIGURE 9-7 *OIM Workflow Designer*

Request-driven provisioning certainly helps us answer the first question, since all user provisioning occurs through a centralized process and is therefore tracking who is being provisioned where. However, for the second question, the request-driven style is not taking responsibility for ensuring if a user *should* access a certain resource, since the provisioning occurs in a discretionary manner. To address this issue, corporate security has to lend a hand by providing us a set of access policies that define rules regarding "who should access what." Once those policies are defined, you can implement them very easily in OIM through the web administrative console's Access Policies section.

The following high-level steps are required to set up an access policy:

1. Go to the Create Access Policy section in the OIM administrative console.
2. Select the resource(s) to be provisioned under the chosen access policy, as shown in Figure 9-8.
3. Set the date this for which access needs to be issued.
4. Select the resource(s) that should be denied to the user through this access policy.
5. Select the user groups that apply to this access policy, as shown in Figure 9-9.

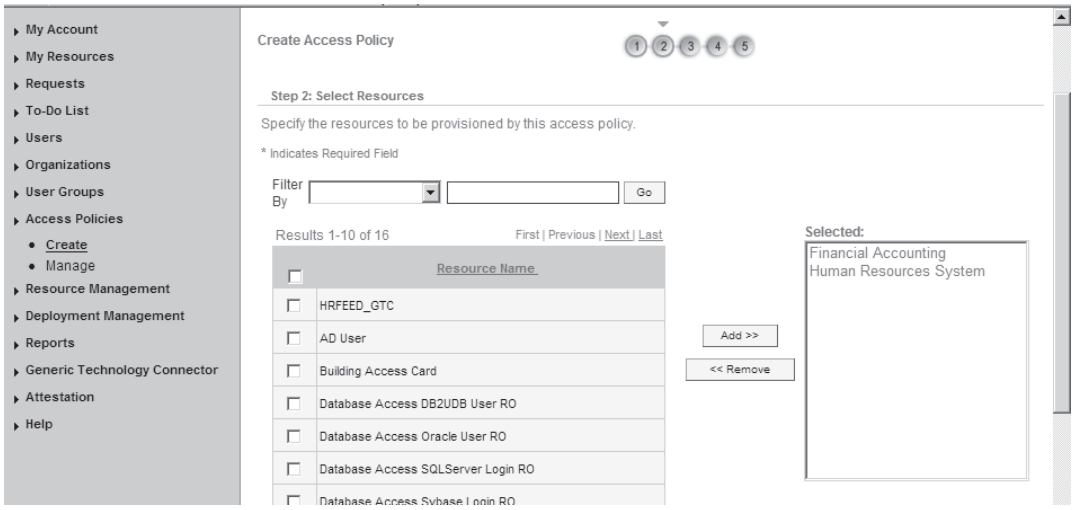


FIGURE 9-8 *Resource selection during access policy configuration*

Once you have defined these four facets of the access policy (what is provisioned, when it is issued, what not to be provisioned, and who this is for), you are ready to automate the majority of your enterprise user provisioning through a collection of these access policies. If you have defined the approval workflows, the access policies will automatically trigger those flows to be routed through the appropriate authorities.

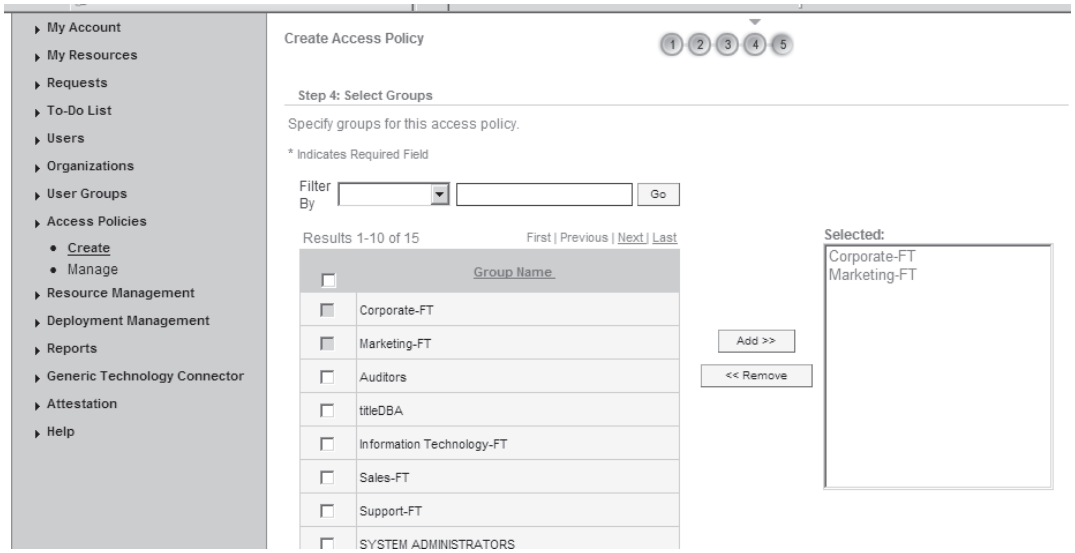


FIGURE 9-9 *User group selection during access policy configuration*

User Provisioning Integrations

One of the key strengths of OIM is the flexibility of its integration platform. However, highly flexible frameworks can often become complex and less usable. As a result, since version 9.1, OIM offers several interaction patterns that allow a user to choose the level of flexibility and sophistication of developing integrations with external systems. I have found that this approach is driven more or less by the 80/20 rule: approximately 80 percent of the use cases are satisfied by 20 percent of the integration types. Those 20 percent integration types are simplified into standard connectors and templates.

Every choice of integration between OIM and an external target systems falls into one of the following categories:

- **Prebuilt connectors** A specific connector implementation for a specific system or application (such as Active Directory, PeopleSoft, SAP, DB2, Oracle Database, and so on).
- **Generic Technology Connector** A connector for commonly-used formats and industry standards (such as flat files, Web Services, and Service Provisioning Markup Language).

Prebuilt Connectors

OIM provides a connector pack that bundles prebuilt and packaged connectors to most third-party systems of all types, including databases, enterprise resource planning (ERP) applications, operating systems, Lightweight Directory Access Protocol (LDAP) servers, and so on. Setting up these connectors in OIM is a fairly straightforward process:

1. Copy the connector files to the OIM server.
2. Import the connector's (XML-based) descriptor file into the OIM repository through the Deployment Manager section in the OIM web console.
3. Define the IT resources associated to this connector,

Through this connector install process, OIM automatically creates the foundational elements of the new resource by creating the necessary resource, IT resource(s), and IT resource type objects associated to the connector. At this point, the environment is ready for basic request-driven provisioning. (See "Discretionary Account Provisioning.")

Generic Technology Connector

One of the first additions Oracle made to the OIM product after its acquisition was the development of the Generic Technology Connector (GTC). Oracle realized that OIM had great capabilities for supporting high-end system integration challenges such as connecting to ERP systems and LDAP servers using prebuilt connectors or developing custom connectors on top of the OIM development framework. However, there was no easy way to perform quick and simple integrations from OIM to smaller scale and perhaps more departmental applications that were built using simpler database technologies such as Application Express or Microsoft Access. As enterprises are looking to automate provisioning to all types of applications (enterprise and departmental), Oracle needed a solution that targeted those applications and systems with a simpler approach to provisioning. This was the genesis of the GTC, introduced in OIM 9.1.

The GTC supports simple integrations to custom-built applications or other systems that rely on simpler data exchange formats such as comma-separated fields. It also supports many industry-standard protocols such as Service Provisioning Markup Language (SPML). The GTC is another example of a packaged integration used for a common set of applications that can read and

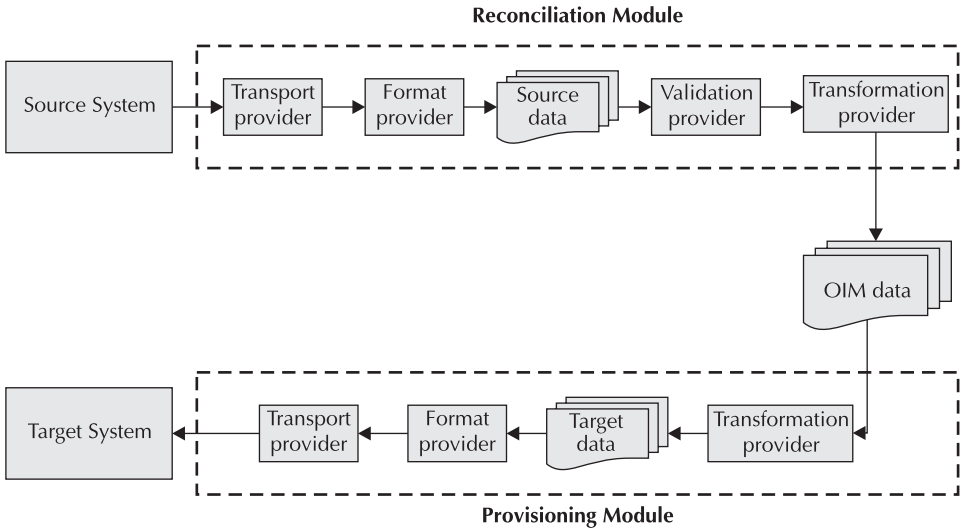


FIGURE 9-10 *The GTC lifecyle*

exchange information in a standard format. While the GTC does not necessarily solve complex integration scenarios, it does provide a quick integration to medium- to low-complexity applications. Figure 9-10 illustrates the provisioning lifecycle of a GTC-based integration.

A GTC-based integration provides a set of packaged functionalities, known as "providers," to perform the different types of actions needed to execute an end-to-end user provisioning process. The process runs starting from identity data reconciliation from a source system to provisioning to a target application.

The GTC is a useful choice whenever you're dealing with applications that can support simpler or standard data exchange formats, such as comma-separated files or the SPML format. The typical cost to set up and maintain a GTC-based integration is much lower than that of other types of OIM integrations. Unlike the prebuilt connectors, the GTC code is shipped with the OIM server so there is no need to install additional software.

Reconciliation Integrations

Two types of system integrations are supported by OIM: *provisioning* and *reconciliation*. Provisioning automates account creation from the OIM server to an application or resource using the data from the OIM repository. Reconciliation automates the creation of an OIM identity record based on an external source of identities (that is, a source of truth). Most often, OIM reconciles from an external human resources application as an authoritative source of employee data and then provisions to business productivity applications, such as email, intranet portals, and other ERP systems.

Reconciliation is often driven by business events such as new hires, new customers, organizational changes, employee transfers, and so on. Since these business events are initiated in an ERP system, most often the Human Resources (HR) system, it makes sense to configure OIM to setup reconciliation with those systems so it can listen for relevant identity events. OIM uses two reconciliation styles: *trusted source reconciliation* and *target resource reconciliation*.

Trusted Source Reconciliation

Trusted source reconciliation (TSR) is used for reconciling information from external authoritative sources (such as HR systems, CRM, and so on) that usually result in creating, modifying, or removing users in the OIM local repository. If the appropriate user groups and access policies are configured, the external reconciliation events can trigger provisioning processes that create or change account data in applications and resources where users are provisioned. For instance, a new employee record entered into the HR application could trigger a record creation in OIM (via reconciliation), which then can subsequently trigger provisioning events (via access policies) to create an e-mail account in the MS Exchange e-mail server.

TSR has two implementation forms:

- **Attribute based** Each trusted source is responsible for reconciling one or more attributes of the user. For instance, the HR system can be the authoritative source that owns the first and last name attributes, whereas the enterprise LDAP server can be the authoritative source for the e-mail address attribute.
- **User-type based** Each trusted source is responsible for reconciling a particular user type in OIM. For instance, the HR system can be the trusted source for employees, whereas the CRM system can be the trusted source for customer user types.

Target Resource Reconciliation

Target resource reconciliation (TRR) is used mainly for reconciling changes to already provisioned users. For instance, if someone changes the phone number of a user in Active Directory without going through the OIM management console, OIM can be configured to reconcile those changes using TRR.

TRR is a very powerful feature in OIM since it can not only choose simple attribute changes from external sources, but it can also be used to identify rogue accounts in external systems quickly. If someone tries to create a privileged account in an external resource (such as Active Directory), TRR can detect that potentially harmful account and take any step that you configure. For example, TRR can configure a policy of automatically disabling rogue accounts until an administrator explicitly re-authorizes the access.

TRR is also useful for reconciling list-of-value fields from target systems (such as LDAP groups, roles, and so on) into OIM so that you can map access policies to actual target system roles and groups.

Compliance Solutions

One of the main drivers of enterprise identity management is the notion of knowing and auditing who has access to what resources. In addition to standard access reporting requirements, compliance mandates often require periodic attestation of users' access to critical applications. Manual attestation of access is an expensive and risky process to enforce, so enterprises are looking to products like OIM to help provide attestation.

Attestation

Attestation requires that a defined approval workflow periodically re-authorizes access to sensitive information (typically financial data) that falls within a particular compliance mandate such as the Sarbanes-Oxley Act (SOX). The person with the authority to re-authorize, also known as the *reviewer*, can have a number of relationships with the user(s) being attested. The reviewer can range from user's direct supervisor, to an application administrator, to the chief operating

officer (COO) of the organization. Basically, the reviewer must have the authority and knowledge to answer the question "who should access what resource." This authority can also be delegated to a different reviewer if the first reviewer is unable to answer that question.

OIM provides a fairly simple way of managing attestation by embedding it into the provisioning process framework. In other words, you can wrap any resource with the need for attestation.

Following are the typical steps you need to take to set up an attestation policy that can govern by user type (such as organizations, roles, and so on) and by the resources that require this form of periodic re-authorization:

1. Go the attestation policy manager. In the OIM administrative console, navigate to the Attestation section.
2. Create a new attestation process. Typically it makes good sense to create attestation processes by resource and/or organization.
3. Specify the scope of users for attestation. This lets you partition how you attest different types of users. For instance, the accounting department attestation process could be approved by the user's supervisor, whereas the sales department attestation could be attested by the sales region's vice president.
4. Specify the scope of applications/resources for attestation. This lets you partition your attestation process by different resources. It often makes sense to use the Resource Audit Objective (which is an attribute associated with a resource) to define your attestation process since different audit objectives require different types of attestations.
5. Specify additional attestation process details. Define additional details such as frequency of the attestations, the process owner, and the grace period for completing the process. Keep in mind that a significant delegation of attestations can occur in large organizations, so the grace period should factor that in.

Figure 9-11 shows how a completed attestation process could look.

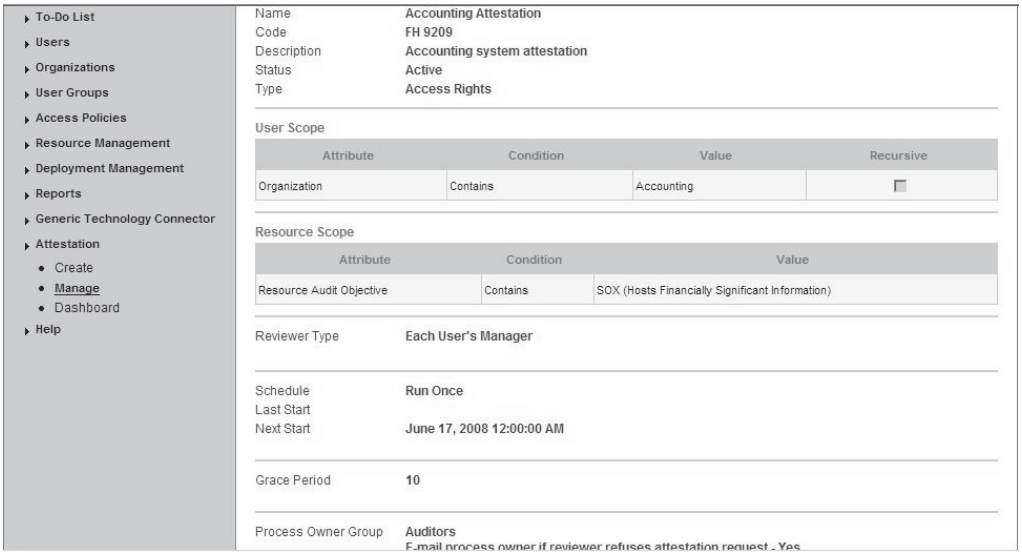


FIGURE 9-11 *Managing an attestation policy in OIM*

Access Reporting

In addition to attestation, another common requirement is to provide compliance and security officers a single consolidated view of who has access to what resources and applications in the enterprise. In other words, officers are looking for a reporting solution around users and their access.

One of the benefits of using a centralized hub-and-spoke architecture (see Chapter 8 for details about this type of architecture) as implemented by a product like OIM is the ability to centralize the identity data into a single repository, which allows you to run reports on top of that data. Access reporting is a key feature of OIM, especially when you need to report on applications that fall under the compliance requirements of SOX and other legislative mandates. Since every new user and modification to existing users is processed through OIM, you can use OIM's reporting infrastructure as a fairly reliable source for asking the question "who *has* access to what" and, occasionally, "who *had* access to what."

Through the web administrative console, OIM offers two types of reporting functionality: *operational* and *historical*. Operational reporting gives the user a snapshot of the current users' access. Historical reporting provides an additional time dimension so that you can view a snapshot in history. A good example that may be driven by SOX requirements is the need to see who had access to the financials application during a certain time period (such as during a corporate quiet period of June 1–30 in 2008). Both types of reports are critical both for compliance and for assuring auditors that the information was under authorized access at all times.

To run either operational or historical reports, navigate to the Reporting section of the OIM administrative UI and click the appropriate report and query with the desired parameters. Figure 9-12 shows a sample report on access to the Financial Accounting application in my environment.

Notice that in Figure 9-12, you can modify parameters to customize your report. You can also export to text-based formats that can be imported into tools such as MS Excel, allowing you to share these reports via e-mail to other parties, such as corporate auditors.

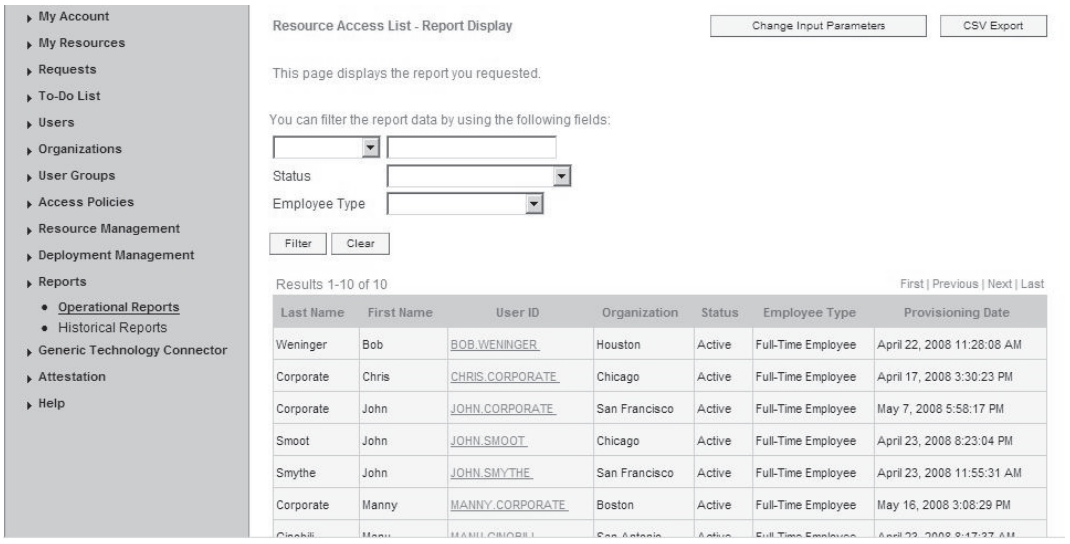


FIGURE 9-12 *Access reporting shows who has access to what and who had access to what.*

OIM Deployment

Every OIM component (design client, web application and core server engine) is written in Java and executes in a multi-tiered deployment model, shown in Figure 9-13.

Client Tier When working with OIM, two types of clients are used: a web-based administrative console and a design-time client. The web administrative console is used mainly for managing users, resources, and all the constructs supporting them. The design-time client is used by the developers of the identity management processes for designing and configuring the core components such as resource objects, IT resources, provisioning processes, and the integration configurations to communicate with the physical applications being provisioned or reconciled. Both types of clients follow a distributed communication model so that you can have many clients from many computers communicating with the same set of policies and objects defined in the OIM business logic tier.

Web Tier This tier exists as a web application container for the OIM administrative user interface. It is a pure Java-based web application environment that uses technologies such as JSP, servlets, Struts, and JavaBeans. By using these standard technologies, the OIM web tier can be deployed in a number of application servers and containers.

Business Logic Tier This tier is the core of the OIM product. In this tier, OIM decides who (the user) to provision where (target resource) and how (the process). This tier is written exclusively in Java and leverages a J2EE design pattern and therefore inherits the core benefits of that combination—platform-neutrality and distributed component architecture. A Java-based

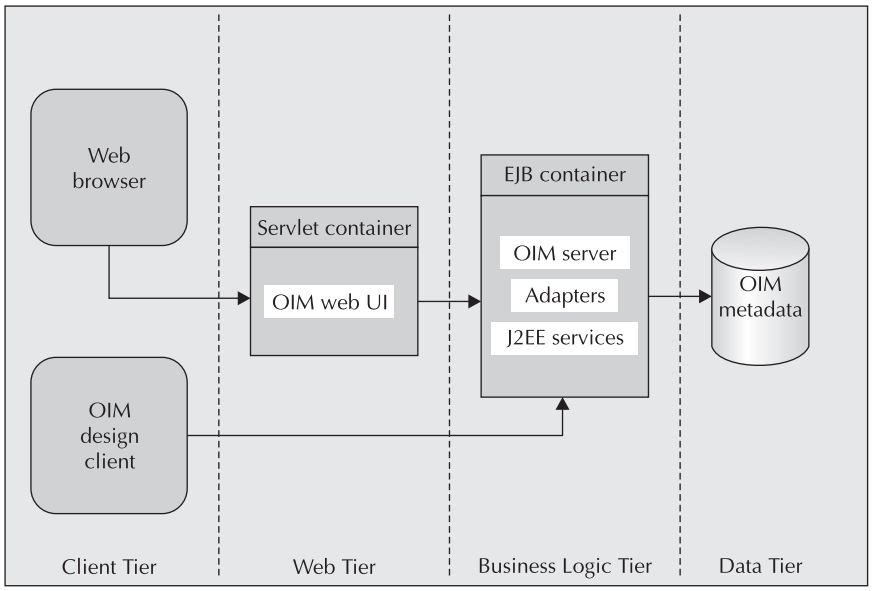


FIGURE 9-13 *OIM deployment architecture*

OIM business tier allows a standard development platform for new integration connectors and adapters. The distributed nature of J2EE allows for the business logic tier to be spread across multiple application server deployments while accessing the common metadata from the data tier.

Data Tier The data tier is a SQL-based relational database that stores all metadata about the identities, accesses, and configurations for the user provisioning platform. The only OIM data that lives outside the database are the JAR (Java Archive) files containing the code to connect to third-party resources and target systems. The data tier is accessed exclusively by the OIM business tier and should not be integrated with any external clients and tools for direct data manipulation. In fact, we recommend that you consider using Oracle database protection technologies, such as Oracle Database Vault and Transparent Data Encryption, to secure and protect the sensitive identity-related metadata stored in the OIM repository. Refer to earlier chapters on TDE and Database Vault for details on how to secure the OIM metadata repository.

Summary

This chapter reviewed the Oracle Identity Manager that addresses the simple to understand but hard to implement area of user provisioning. Provisioning is a mandatory process inside every enterprise, executing constantly either in a manual or an automated manner. As a result, optimizing the processes around provisioning is critical to both achieve operational efficiency and deliver assurance that access policies are not being violated or ignored. Security issues include orphaned accounts that are not de-provisioned. Open, unused accounts are footholds for disgruntled employees and attackers and are at the top of the list of things that compliance auditors look for. As a result, a truly successful user provisioning solution balances building better optimized processes and policies to lower administrative burden with instituting consistency of identity management, in terms of the way it grants and monitors access to information, to result in a higher level of security and protection of all enterprise assets.