

Oracle® Role Manager

Developer's Guide

Release 10g (10.1.4.2)

E14614-02

July 2009

Oracle Role Manager Developer's Guide Release 10g (10.1.4.2)

E14614-02

Copyright © 2008, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: Carla Fabrizio

Contributor: Miles Chaston, Ashish Chugh, April Escamilla, Bennett Falk, Stephen Grenholm, Parvinder Kaur, Avinash Mittal, Devender Sharma

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xi
Audience.....	xi
Documentation Accessibility	xi
Related Documents	xii
Conventions	xii
1 Oracle Role Manager Data Model	
1.1 General Concepts	1-1
1.1.1 Data Model Layering	1-1
1.1.2 Domains and Attributes	1-2
1.1.3 Permissions and System Privileges	1-2
1.2 Roles.....	1-3
1.2.1 abstractRole	1-4
1.2.2 approverRole	1-6
1.2.3 businessRole	1-7
1.2.4 itRole.....	1-9
1.2.5 systemRole.....	1-11
1.2.6 roleGrant	1-13
1.2.7 roleMapping	1-13
1.2.8 relevantRoleAttribute.....	1-13
1.2.9 userRoleAssignment	1-14
1.3 Users	1-14
1.3.1 abstractIdentity	1-15
1.3.2 person	1-16
1.3.3 systemIdentity.....	1-19
1.4 Organizations	1-21
1.4.1 abstractOrg	1-22
1.4.2 building	1-24
1.4.3 country	1-24
1.4.4 dcObject.....	1-25
1.4.5 division.....	1-26
1.4.6 floor.....	1-26
1.4.7 locality	1-27

1.4.8	organization.....	1-28
1.4.9	ou.....	1-29
1.4.10	room.....	1-30
1.5	Hierarchies	1-31
1.5.1	hierarchy	1-32
1.5.2	reportingHierarchy.....	1-33
1.5.3	costCenterHierarchy.....	1-33
1.5.4	locationHierarchy	1-34
1.6	Internal Security and Access Control.....	1-34
1.6.1	systemPermission	1-35
1.6.2	systemPrivilege	1-36
1.6.3	systemResource.....	1-37
1.6.4	systemResourceType.....	1-38
1.6.5	sysPermissionAssociation	1-39
1.6.6	sysRolePrivilegeMapping.....	1-39
1.7	Entitlements	1-40
1.7.1	itPrivilege	1-40
1.7.2	itRolePrivilegeMapping.....	1-41
1.8	System Infrastructure	1-42
1.8.1	auditEvent.....	1-42
1.8.2	auditEventDetail	1-43
1.8.3	baseBundle.....	1-44
1.8.4	configuration	1-44
1.8.5	localizedBundle.....	1-44
1.8.6	pluginPack	1-45

2 Configuring the Data Model

2.1	Best Practices	2-1
2.2	Viewing the Standard Model Configuration	2-2
2.3	Adding and Modifying Attributes	2-2
2.4	Adding and Modifying Reference Attributes.....	2-4
2.5	Adding Structural Type Objects	2-6
2.6	Extending the Standard Configuration	2-7
2.7	Modifying the Standard Configuration.....	2-8
2.8	Deploying Data Model Customizations	2-9

3 Configuring the User Interface

3.1	Best Practices	3-1
3.2	Extracting Files from the Web Application Archive.....	3-2
3.3	Modifying Appearance and Style.....	3-2
3.3.1	Changing the Header Logo	3-3
3.3.2	Changing the Header Background and Link Colors	3-3
3.4	Modifying the Search Component	3-4
3.5	Deploying UI Customizations.....	3-4

A XML Schema Definitions

A.1	Object Type Definition	A-1
A.2	Model Definition	A-9

Index

List of Examples

2-1	Creating an Attribute	2-3
2-2	Modifying an Attribute	2-3
2-3	Creating a Relationship.....	2-5
2-4	Creating a Structural Type	2-6
3-1	Changing the Logo	3-3
3-2	Changing the Background and Link Colors	3-3
3-3	Changing the Number of Rows in the Search Results.....	3-4

List of Figures

1-1	Role Object Relationship Diagram	1-4
1-2	Users Entity Relationship Diagram.....	1-15
1-3	Organization Model Entity Relationship Diagram.....	1-22
1-4	Hierarchy Model Entity Relationship Diagram	1-32
1-5	Internal Security Entity Relationship Diagram	1-35
1-6	Entitlement Model Entity Relationship Diagram.....	1-40

List of Tables

1-1	System Permissions	1-3
1-2	abstractRole Attributes.....	1-5
1-3	abstractRole Relationship Paths (primordial).....	1-5
1-4	abstractRole Relationship Paths (standard).....	1-6
1-5	abstractRole Related Audit Objects.....	1-6
1-6	approverRole Attributes	1-6
1-7	approverRole Relationship Paths (standard).....	1-7
1-8	businessRole attributes.....	1-8
1-9	businessRole Relationship Paths (primordial)	1-8
1-10	businessRole Relationship Paths (standard).....	1-9
1-11	businessRole Related Audit Objects.....	1-9
1-12	itRole attributes	1-9
1-13	itRole Relationship Paths (primordial)	1-10
1-14	itRole Relationship Paths (standard).....	1-11
1-15	itRole Related Audit Objects	1-11
1-16	systemRole Attributes	1-11
1-17	systemRole Relationship Paths (primordial)	1-12
1-18	systemRole Relationship Paths (standard).....	1-12
1-19	systemRole Related Audit Objects	1-12
1-20	roleGrant Relationship Paths (primordial)	1-13
1-21	roleMapping Relationship Paths (primordial)	1-13
1-22	relevantRoleAttribute Attributes (primordial).....	1-14
1-23	relevantRoleAttribute Relationship Paths (primordial).....	1-14
1-24	userRoleAssignment Relationship Paths (primordial).....	1-14
1-25	abstractIdentity Attributes.....	1-16
1-26	abstractRole Relationship Paths (primordial).....	1-16
1-27	abstractIdentity Related Audit Objects.....	1-16
1-28	person Attributes	1-17
1-29	person Relationship Paths	1-19
1-30	systemIdentity Attributes	1-20
1-31	abstractOrg Attributes.....	1-22
1-32	abstractOrg Relationship Paths (standard)	1-23
1-33	building attributes	1-24
1-34	country attributes.....	1-25
1-35	dcObject attributes	1-25
1-36	division attributes	1-26
1-37	floor attributes	1-27
1-38	locality attributes.....	1-27
1-39	organization attributes	1-28
1-40	ou attributes	1-29
1-41	room attributes	1-30
1-42	abstractOrg Attributes.....	1-32
1-43	hierarchy Relationship Paths (primordial).....	1-33
1-44	reportingHierarchy Node Relationship Paths (standard)	1-33
1-45	reportingHierarchy Node Relationship Paths (standard)	1-34
1-46	locationHierarchy Node Relationship Paths (standard)	1-34
1-47	systemPermission Attributes	1-35
1-48	systemPermission Relationship Paths (primordial)	1-36
1-49	systemPrivilege Attributes	1-36
1-50	systemPrivilege Relationship Paths (primordial)	1-36
1-51	systemPrivilege Related Audit Objects (primordial).....	1-37
1-52	systemResource Attributes.....	1-37
1-53	systemResource Relationship Paths (primordial).....	1-38

1-54	systemResourceType Attributes	1-38
1-55	systemResourceType Relationship Paths (primordial)	1-38
1-56	sysPermissionAssociation Relationship Paths (primordial).....	1-39
1-57	sysRolePrivilegeMapping Relationship Paths (primordial).....	1-39
1-58	itPrivilege Attributes	1-40
1-59	itPrivilege Relationship Paths (primordial)	1-41
1-60	itPrivilege Related Audit Objects (standard).....	1-41
1-61	itPrivilege Relationship Paths (primordial)	1-41
1-62	itPrivilege Related Audit Objects (standard).....	1-41
1-63	auditEvent Attributes (primordial).....	1-42
1-64	Relationship Paths for auditEvent.....	1-43
1-65	auditEventDetail Attributes (primordial)	1-43
1-66	Relationship Paths for auditEventDetail (primordial)	1-44
2-1	Reference Attribute Components	2-4

Preface

The *Oracle Role Manager Developer's Guide* describes the Oracle Role Manager data model and information for customizing the application data model and user interface.

Audience

This document is intended for those who are involved in the development and extension of Oracle Role Manager.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Related Documents

For more information, refer to the following documents:

- *Oracle Role Manager Release Notes*
- *Oracle Role Manager User's Guide*
- *Oracle Role Manager Installation Guide*
- *Oracle Role Manager Integration Guide*
- *Oracle Role Manager Administrator's Guide*
- *Oracle Role Manager Java API Reference*
- *Oracle Role Manager Licensing Information*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Oracle Role Manager Data Model

This chapter describes data model for Oracle Role Manager. This document contains information on the primordial and standard data models and includes the following aspects:

- [General Concepts](#)
- [Roles](#)
- [Users](#)
- [Organizations](#)
- [Hierarchies](#)
- [Internal Security and Access Control](#)
- [Entitlements](#)
- [System Infrastructure](#)

1.1 General Concepts

The general concepts described in this section reflect the important information about the Oracle Role Manager data model and how the application is built with that model.

- [Data Model Layering](#)
- [Domains and Attributes](#)
- [Permissions and System Privileges](#)

1.1.1 Data Model Layering

The Oracle Role Manager data model follows a multi-tier approach as follows:

- The base data model objects that are required for Oracle Role Manager to function as designed are defined in the *primordial model*. The primordial model is loaded as part of the initial deployment of Oracle Role Manager.
- The *standard model* provides extensions to the primordial layer, building on base objects. The objects in the standard model are required for the Web application to function as designed. This should be considered when customizing the model-related aspects of the user interface. For example, one customization approach could be satisfied by simply extending the standard model where another approach could require the standard model to be altered or even not deployed at all.

The standard model is defined in a file named `standard.xml`. To view the content of this file, refer to [Section 2.2](#).

- Optional extensions, such as the *oim_integration* model that supports the Oracle Role Manager Integration Library with Oracle Identity Manager, contain extensions to the standard model.

The Oracle Role Manager *application data model* (extensions to the primordial model) is configurable and supports incremental model additions and customizations. For example, the model containing the extensions required to use the Oracle Role Manager Integration Library with Oracle Identity Manager can be deployed after Oracle Role Manager is installed and in production. Incremental data model deployments can be performed while the application and database are running without disrupting the environment.

One effect of this layered approach is that things defined on a lower layer have very specific ways that they can be extended. For example:

- A domain definition (attribute definition) in one configuration cannot be changed or removed in another. For example, if there is a domain defined in `standard.xml`, you cannot deploy another data model definition at the same time that alters or removes that domain. (Refer to [Section 1.1.2](#) for more information about domains.)
- An object type defined in one configuration cannot be changed or removed in another.
- Hierarchies can only be defined once and not extended in another configuration.

1.1.2 Domains and Attributes

The Oracle Role Manager model uses the concept of *domains*. Domains are attributes (values on objects) within the data model and are defined individually before they are associated with object types.

A domain defines the data type and some inherent aspects of that domain such as various aspects (maximum length, size, precision). Supported data types are String (CLOB), integer, decimal, datetime, Boolean, binary (BLOB), and reference (Long).

The domain can then be associated to object types to define attributes. When associated to those object types, further constraints can be added, such as string pattern constraints, enumerated value lists and value ranges.

Note: Constraints defined in the domain definition, not the associated object type, cannot be altered. The same is true for domain constraints defined in the primordial model.

1.1.3 Permissions and System Privileges

The internal permissions and privileges in Oracle Role Manager are key to the system's internal security and access model and give users the access to types of events in Oracle Role Manager. These permissions and privileges should not be confused with the external privileges (entitlements) that are used to represent external security models.

Permissions represent general actions that a user may want to do with objects in Oracle Role Manager. These are defined globally and associated with object types in the *standard permissions* model. The standard permissions model is defined in the `standard_permissions.xml` file. To view the content of this file, refer to [Section 2.2](#).

Note: Permissions are not inherited. For example, `person` does not inherit the permissions defined for `abstractIdentity`.

System privileges are the association of a permission and an object type, such as "audit person" or "manage organization." System privileges also support inheritance, for example, a permission on `abstractOrg` is a general case of the same permission bound to the `organization` sub-type. Using this example, if a person has the `manage` permission on `abstractOrg` but not on `organization`, that person would implicitly have the `manage` permission on `organization` because `organization` extends `abstractOrg`.

In addition to defining permissions and system privileges, the data model is also the place where particular security policy enforcement is defined. Security policy enforcement is done through policies on an object type that specify which system privilege is necessary to *manage* (modify) an object of that type and also which system privilege is necessary to *read* the audit details of objects of that type.

The standard permissions model extends the `person` object type to include the `audit`, `delegate`, `grant`, and `manage` permissions on `person` objects. The permissions in the Oracle Role Manager system are defined in both the primordial model and the standard model as shown in [Table 1-1](#).

Table 1-1 System Permissions

Permission ID	Defined in	Default Title	Description
any	primordial	All	Have all permissions for an object.
audit	standard_permissions	Audit	Have audit permission for an object (Access through the Web application's user interface to transaction history.)
delegate	standard_permissions	Delegate	Have delegate permission for an object. (Used for role delegation.)
grant	standard_permissions	Grant	Have grant permission for an object. (Used for role objects for user-role assignments.)
manage	standard_permissions	Manage	Have management permission for an object (Includes create, update, and delete.)

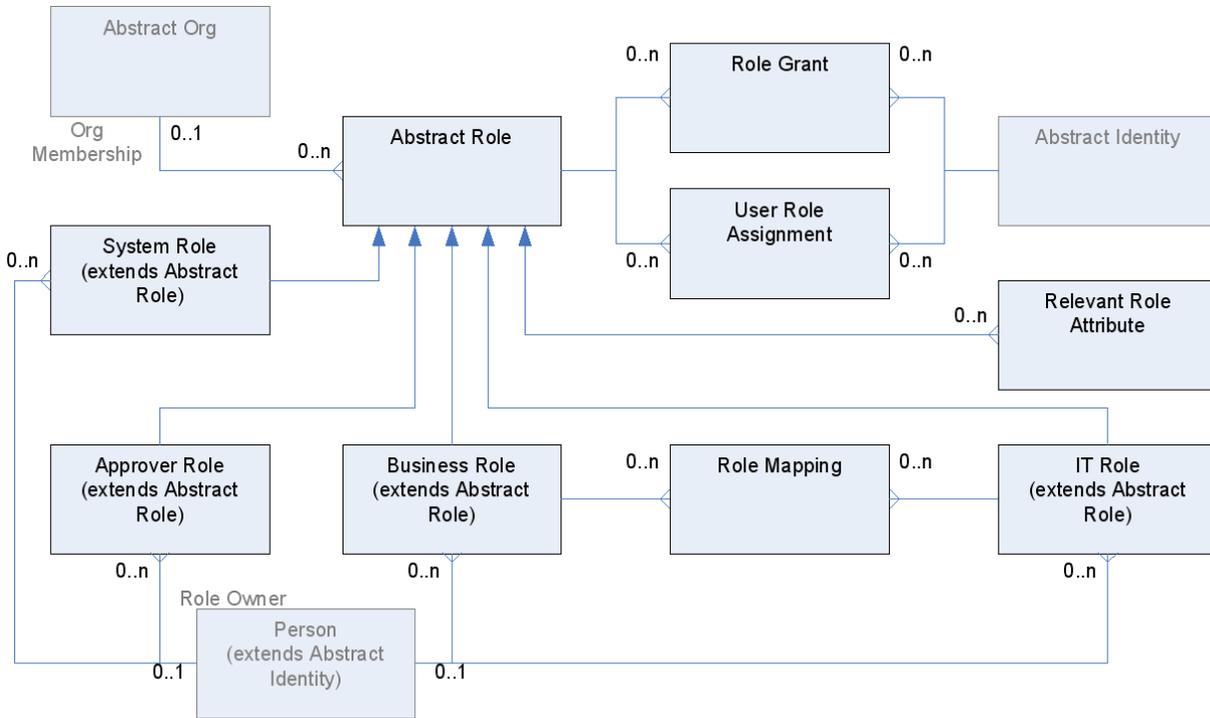
1.2 Roles

This section describes all of the role objects and role-related relationships in the data model, illustrated in [Figure 1-1](#). The following objects are described in this section:

- [abstractRole](#)
- [approverRole](#)
- [businessRole](#)
- [itRole](#)
- [systemRole](#)
- [roleGrant](#)
- [roleMapping](#)

- `relevantRoleAttribute`
- `userRoleAssignment`

Figure 1–1 Role Object Relationship Diagram



1.2.1 abstractRole

The `abstractRole` object type is the abstract base type from which all roles are extended. The use of an abstract base type is necessary to permit the generalization patterns of role grants, user role assignments and general role functionality.

An instance of `abstractRole` is never explicitly created, but rather, is created as part of creating instances of its subtypes. `abstractRole` is related to `abstractOrg` to represent its membership in the reporting hierarchy for administrative purposes.

The base definition of `abstractRole` is in the primordial model and it is extended in the standard model. `abstractRole` has no permissions as they are defined on its subtypes. The external title for `abstractRole` instances is "all role," which is used in tokenized system privileges names, for example, "Audit all role objects" and "Manage all role objects."

1.2.1.1 Attributes for abstractRole

The attributes for abstract role, shown in [Table 1–2](#) are defined in both the primordial model and the standard model.

Table 1–2 *abstractRole Attributes*

Attribute ID	Defined in	Type	Default Title	Constraints
description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
displayName	primordial	String	Display name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
eligibilityRule	primordial	CLOB	Eligibility rule	max-length=100000000
membershipRule	primordial	CLOB	Membership rule	max-length=100000000
roleType	primordial	String	Role type	Must not be null. Allowable values are <i>static</i> or <i>dynamic</i> . max-length=32
simpleDynamic	primordial	Boolean	Simple Dynamic	
status	primordial	String	Status	Must not be null. Allowable values are <i>inactive</i> or <i>active</i> . Default is <i>active</i> . max-length=32
uniqueName	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.2.1.2 Relationship Paths for abstractRole (primordial)

The relationship paths shown in [Table 1–3](#), are defined in the primordial model.

Table 1–3 *abstractRole Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
socHierarchy	socHierarchyRoles	many-to-one	hierarchy
role_grants	role	one-to-many	roleGrant
grantees	granted_roles	many-to-many	abstractIdentity
user_role_assignments	role	one-to-many	userRoleAssignment
assignees	assigned_roles	many-to-many	abstractIdentity

1.2.1.3 Relationship Paths for abstractRole (standard)

The relationship paths shown in [Table 1–4](#) are defined in the standard model.

Table 1–4 abstractRole Relationship Paths (standard)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
parent_reporting_organization	child_reporting_roles	many-to-one	abstractOrg
parent_cost_center_organization	child_cost_center_roles	many-to-one	abstractOrg
parent_location_organization	child_location_roles	many-to-one	abstractOrg

1.2.1.4 Related audit objects for abstractRole

The related audit object for `abstractRole`, shown in [Table 1–5](#), is defined in the standard model.

Table 1–5 abstractRole Related Audit Objects

Related Object Type	Incoming Relationship Path
roleGrant	role
userRoleAssignment	role

1.2.2 approverRole

The `approverRole` subtype of `abstractRole` is specifically for representing approver roles. It supports only being a dynamic role to fit the recommended model of dynamically managed role memberships. The special aspect of approver roles is that the membership rule resolves in a relative as opposed to a simple fashion, meaning that the set of members of the role can be determined by the approval context object provided when membership is resolved. When defining rules for approver roles, any object type can be provided as an approval context object to resolve the set of people who qualify as approvers.

The base definition of `approverRole` is in the primordial model and it is extended in the standard model. `approverRole` has the `manage`, and `audit` permissions. The external title for `approverRole` is "Approver Role," which is used in tokenized system privilege names, for example "Audit Approver Role objects."

For more general information about Approver Roles and examples of eligibility rules, refer to the *Oracle Role Manager User's Guide*.

1.2.2.1 Attributes for approverRole

The attributes for `approverRole`, shown in [Table 1–6](#) are defined in the primordial model. All attributes are inherited from the `abstractRole` object type from which `approverRole` is extended.

Table 1–6 approverRole Attributes

Inherited Attribute ID	Defined in	Type	Default Title	Constraints
description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
displayName	primordial	String	Display name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256

Table 1–6 (Cont.) approverRole Attributes

Inherited Attribute ID	Defined in	Type	Default Title	Constraints
eligibilityRule	primordial	CLOB	Eligibility rule	max-length=100000000
membershipRule	primordial	CLOB	Membership rule	max-length=100000000
roleType	primordial	String	Role type	Must not be null. Allowable values are static or dynamic. max-length=32
simpleDynamic	primordial	Boolean	Simple Dynamic	
status	primordial	String	Status	Must not be null. Allowable values are inactive or active. Default is active. max-length=32
uniqueName	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.2.2.2 Relationship Paths for approverRole (standard)

The relationship path shown in [Table 1–7](#), is defined in the standard model.

Table 1–7 approverRole Relationship Paths (standard)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
roleOwner	ownedApproverRoles	many-to-one	person

1.2.3 businessRole

The `businessRole` subtype of `abstractRole` is specifically for representing business roles. It supports being either a static or a dynamic role. Static grants of business roles support sphere of control within the Oracle Role Manager system. (Refer to [Section 1.5](#) for information about sphere of control.)

The base definition of `businessRole` is in the primordial model and it is extended in the standard model. `businessRole` has the `manage`, `audit`, `grant`, and `delegate` permissions. The external title for `businessRole` is "Business Role," which is used in tokenized system privilege names, for example "Grant Business Role objects."

For more general information about Business Roles and examples of membership rules, refer to the *Oracle Role Manager User's Guide*.

1.2.3.1 Attributes for businessRole

The attributes for `businessRole`, shown in [Table 1–8](#) are defined in the primordial model and extended in the standard model. Many attributes are inherited from the `abstractRole` object type from which `businessRole` is extended.

Table 1–8 *businessRole* attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
approverType		primordial	String	Approver	Must not be null. Allowable values are noapproval, roleowner, or specificperson. Default is noapproval.
isDelegatable		primordial	Boolean	Can Be Delegated	Default is false.
responsibility		standard	String	Responsibilities	max-length=4000
	description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	primordial	String	Display name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	eligibilityRule	primordial	CLOB	Eligibility rule	max-length=100000000
	membershipRule	primordial	CLOB	Membership rule	max-length=100000000
	roleType	primordial	String	Role type	Must not be null. Allowable values are static or dynamic. max-length=32
	simpleDynamic	primordial	Boolean	Simple Dynamic	
	status	primordial	String	Status	Must not be null. Allowable values are inactive or active. Default is active. max-length=32
	uniqueName	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.2.3.2 Relationship Paths for businessRole (primordial)

The relationship paths shown in [Table 1–9](#), are defined in the primordial model.

Table 1–9 *businessRole* Relationship Paths (primordial)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
business_to_it_role_mappings	business_role	one-to-many	roleMapping
it_roles	business_roles	many-to-many	itRole

Refer to [Table 1–3](#) for additional relationship paths of abstractRole, which businessRole extends.

1.2.3.3 Relationship Paths for businessRole (standard)

The relationship path shown in [Table 1–10](#), is defined in the standard model.

Table 1–10 businessRole Relationship Paths (standard)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
approver	approverBusinessRoles	one-to-many	person
roleOwner	ownedBusinessRoles	many-to-one	person

1.2.3.4 Related audit objects for businessRole

The related audit object for `businessRole`, shown in [Table 1–11](#), is defined in the standard model.

Table 1–11 businessRole Related Audit Objects

Related Object Type	Incoming Relationship Path
roleMapping	business_role

1.2.4 itRole

The `itRole` subtype of `abstractRole` is specifically for representing IT roles. It is constrained to only be a static role, but does support mapping to zero or more business roles, representing membership inheritance (in other words, all people who have the business roles mapped to this IT role are considered members of the IT role).

The base definition of `itRole` is in the primordial model and it is extended in the standard model. `itRole` has the `manage`, `audit`, `grant`, and `delegate` permissions. The external title for `itRole` instances is "IT Role," which is used in tokenized system privilege names, for example "Audit IT Role objects."

For more general information about IT Roles and examples of membership rules, refer to the *Oracle Role Manager User's Guide*.

1.2.4.1 Attributes for itRole

The attributes for `itRole`, shown in [Table 1–12](#) are defined in both the primordial model and the standard model. Many attributes are inherited from the `abstractRole` object type from which `itRole` is extended.

Table 1–12 itRole attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
isDelegatable		primordial	Boolean	Can Be Delegated	Default is false.
isFinanceRelated		standard	Boolean	Finance related	Default is false.
isHighRisk		standard	Boolean	High Risk	Default is false.
isNpiRelated		standard	Boolean	Non-Public Personal Information Related	Default is false.
isSoxRelated		standard	Boolean	Sarbanes-Oxley Related	Default is false.
	description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000

Table 1–12 (Cont.) itRole attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
	displayName	primordial	String	Display name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	eligibilityRule	primordial	CLOB	Eligibility rule	max-length=100000000
	membershipRule	primordial	CLOB	Membership rule	max-length=100000000
	roleType	primordial	String	Role type	Must not be null. Allowable values are static or dynamic. max-length=32
	simpleDynamic	primordial	Boolean	Simple Dynamic	
	status	primordial	String	Status	Must not be null. Allowable values are inactive or active. Default is active. max-length=32
	uniqueName	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=256

1.2.4.2 Relationship Paths for itRole (primordial)

The relationship paths shown in [Table 1–13](#), are defined in the primordial model.

Table 1–13 itRole Relationship Paths (primordial)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
business_to_it_role_mappings	it_role	one-to-many	roleMapping
business_roles	it_roles	many-to-many	businessRole
it_role_privilege_mappings	it_role	one-to-many	itRolePrivilegeMapping
it_privileges	it_roles	many-to-many	itPrivilege

Refer to [Table 1–3](#) for additional relationship paths of abstractRole, which itRole extends.

1.2.4.3 Relationship Paths for itRole (standard)

The relationship path shown in [Table 1–14](#), is defined in the standard model.

Table 1–14 *itRole Relationship Paths (standard)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
roleOwner	ownedITRoles	many-to-one	person

1.2.4.4 Related audit objects for itRole

The related audit objects for `itRole`, shown in [Table 1–15](#), are defined in the standard model.

Table 1–15 *itRole Related Audit Objects*

Related Object Type	Incoming Relationship Path
roleMapping	it_role
itRolePrivilegeMapping	it_role

1.2.5 systemRole

The `systemRole` subtype of `abstractRole` is specifically for representing system roles. It only supports being a static role. Static grants of system roles support sphere of control by defining the objects on which the grantee has the associated privileges to act upon. For more information about privileges, refer to [Section 1.7](#). For more information about sphere of control, refer to [Section 1.5](#).

The base definition of `systemRole` is in the primordial model and it is extended in the standard model. `systemRole` has the `manage`, `audit`, and `grant` permissions. The external title for `systemRole` is "System Role," which is used in tokenized system privilege names, for example "Grant System Role objects."

1.2.5.1 Attributes for systemRole

The attributes for `systemRole`, shown in [Table 1–16](#) are defined in both the primordial model and the standard model. Many attributes are inherited from the `abstractRole` object type from which `systemRole` is extended.

Table 1–16 *systemRole Attributes*

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
isDelegatable		primordial	Boolean	Can Be Delegated	Default is <code>false</code> .
	description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	primordial	String	Display name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	eligibilityRule	primordial	CLOB	Eligibility rule	max-length=10000000
	membershipRule	primordial	CLOB	Membership rule	max-length=10000000

Table 1–16 (Cont.) systemRole Attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
	roleType	primordial	String	Role type	Must not be null. Allowable values are static or dynamic. max-length=32
	simpleDynamic	primordial	Boolean	Simple Dynamic	
	status	primordial	String	Status	Must not be null. Allowable values are inactive or active. Default is active. max-length=32
	uniqueName	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.2.5.2 Relationship Paths for systemRole (primordial)

The relationship paths shown in [Table 1–17](#), are defined in the primordial model.

Table 1–17 systemRole Relationship Paths (primordial)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
sys_role_privilege_mappings	system_role	one-to-many	sysRolePrivilegeMapping
system_privileges	system_roles	many-to-many	systemPrivilege

Refer to [Table 1–3](#) for additional relationship paths of `abstractRole`, which `systemRole` extends.

1.2.5.3 Relationship Paths for systemRole (standard)

The relationship path shown in [Table 1–18](#), is defined in the standard model.

Table 1–18 systemRole Relationship Paths (standard)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
roleOwner	ownedSystemRoles	many-to-one	person

1.2.5.4 Related audit objects for systemRole

The related audit object for `systemRole`, shown in [Table 1–19](#), is defined in the standard model.

Table 1–19 systemRole Related Audit Objects

Related Object Type	Incoming Relationship Path
sysRolePrivilegeMapping	system_role

1.2.6 roleGrant

The `roleGrant` relationship object type represents an explicit membership of an abstract identity to a role. Due to its relationship nature, there can only be one grant per abstract identity per role. Multiple grants that differentiate themselves by sphere of control are represented by zero or more relationships between this object and the sphere of control (represented by hierarchy index mix-ins).

To represent delegated grants, the grantee from which the delegated grant originates is stored in the `originalGrantee_id` reference attribute. If the grant was due to delegation, this attribute is null. Refer to [Section 1.5](#) for more information about sphere of control on role grants.

The base definition of `roleGrant` is in the primordial model and it is extended in the standard model. The external title for `roleGrant` is "Role Grant," which is used in system messages that display in the log files and application server console.

1.2.6.1 Relationship Paths for roleGrant (primordial)

The relationship paths shown in [Table 1–20](#), are defined in the primordial model.

Table 1–20 *roleGrant Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
originalGrantee	delegatedRoleGrants	many-to-one	abstractIdentity
grantee	role_grants	many-to-one	abstractIdentity
role	role_grants	many-to-one	abstractRole

1.2.7 roleMapping

The `roleMapping` relationship object type represents the mapping between a business role and an IT role. The business ramifications of this are that the effective membership of the IT role is the union of members of all of its mapped business roles. Members of the mapped business role have, as a result of role mapping, any entitlements associated with the mapped IT role. Refer to [Section 1.7](#) for information about the mapped entitlements.

The base definition of `roleMapping` is in the primordial model and it is extended in the standard model. The external title for `roleMapping` is "Business Role to IT Role Mapping," which is used in system messages that display in the log files and application server console.

1.2.7.1 Relationship Paths for roleMapping (primordial)

The relationship paths shown in [Table 1–21](#), are defined in the primordial model.

Table 1–21 *roleMapping Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
business_role	business_to_it_role_mappings	many-to-one	businessRole
it_role	business_to_it_role_mappings	many-to-one	itRole

1.2.8 relevantRoleAttribute

The `relevantRoleAttribute` structural type object stores attributes that are relevant to the type of role and is used to enable a fast search to find "which simple dynamic roles would be affected by a particular set of attribute values changing." For example,

to resolve to all person objects whose first name is John, the `relevantRoleAttribute` table contains the `givenName` attribute.

This type enables fast synchronization of user role assignments when a system identity's attributes are modified. The external title for `relevantRoleAttribute` is "Relevant Role Attribute," which is used in system messages that display in the log files and application server console.

1.2.8.1 Attributes for `relevantRoleAttribute`

The attribute for `relevantRoleAttribute`, shown in [Table 1–22](#) is defined in the primordial model.

Table 1–22 *relevantRoleAttribute Attributes (primordial)*

Attribute ID	Defined in	Type	Default Title	Constraints
<code>attributeName</code>	primordial	String	Attribute Name	Must not be null. max-length=64

1.2.8.2 Relationship Paths for `relevantRoleAttribute` (primordial)

The relationship path shown in [Table 1–23](#), is defined in the primordial model.

Table 1–23 *relevantRoleAttribute Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
<code>role</code>	<code>relevant_role_attributes</code>	many-to-one	<code>abstractRole</code>

1.2.9 `userRoleAssignment`

The `userRoleAssignment` relationship object type represents a derived (implicit) membership of an abstract identity to a role (subtype of `abstractRole`). These objects are created and deleted by the framework based on the results of a role's membership rule, thus representing the derived version of a role grant. Due to its relationship nature, there can only be one assignment per abstract identity per role. Unlike `roleGrant` objects, `userRoleAssignment` objects do not support concepts of sphere of control or delegation.

The definition of `userRoleAssignment` is in the primordial model. The external title for `userRoleAssignment` is "User Role Assignment," which is used in system messages that display in the log files and application server console.

1.2.9.1 Relationship Paths for `userRoleAssignment` (primordial)

The relationship path shown in [Table 1–24](#), is defined in the primordial model.

Table 1–24 *userRoleAssignment Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
<code>assignee</code>	<code>user_role_assignments</code>	many-to-one	<code>abstractIdentity</code>
<code>role</code>	<code>user_role_assignments</code>	many-to-one	<code>abstractRole</code>

1.3 Users

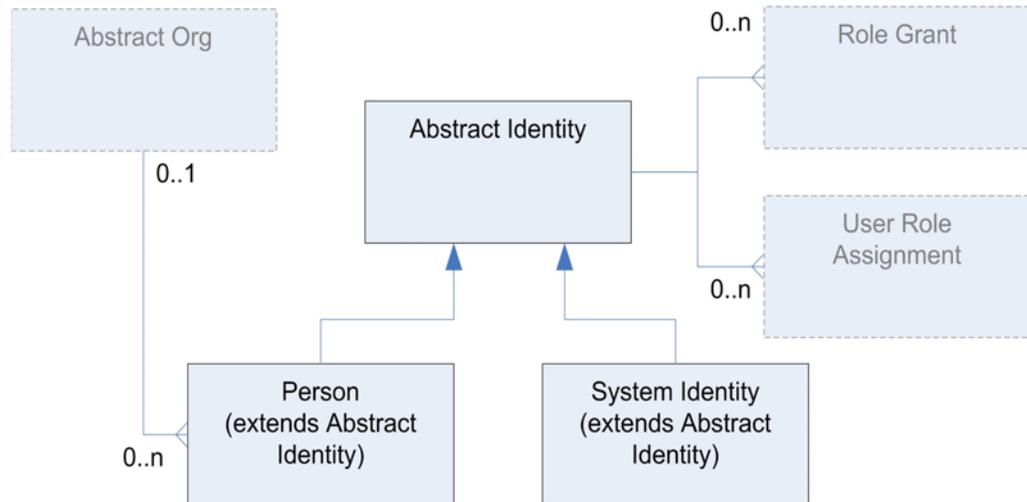
The objects in this section represent things that can interact with Oracle Role Manager, such as application users, system users, or integrated users, as illustrated in

Figure 1–2. Whether a particular object can be used to interact with the system depends on the ability for it to authenticate.

The following objects are described in this section:

- `abstractIdentity`
- `person`
- `systemIdentity`

Figure 1–2 Users Entity Relationship Diagram



The object types shown in the diagram with solid borders are described in the following sections.

1.3.1 `abstractIdentity`

The `abstractIdentity` object type is the abstract root type for all users. This object type is necessary to allow general, user-oriented operations like role grants, user role assignments and authorization. With this base object type, there is shared business logic for identities and people and the ability to define its subtypes independently.

An instance of `abstractIdentity` is never explicitly created, but rather, is created as part of creating instances of its subtypes.

The base definition of `abstractIdentity` is in the primordial model and it is extended in the standard model. `abstractIdentity` has no permissions as they are defined on its subtypes. The external title for `abstractIdentity` instances is "User," which is used in system messages that display in the log files and application server console.

1.3.1.1 Attributes for `abstractIdentity`

The attributes for `abstractIdentity`, shown in Table 1–25 are defined in both the primordial model and the standard model.

Table 1–25 *abstractIdentity Attributes*

Attribute ID	Defined in	Type	Default Title	Constraints
displayName	primordial	String	Display Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
locale	primordial	String	Locale	max-length=128
status	primordial	String	Status	Must not be null. Allowable values are inactive or active. Default is inactive. max-length=32
uniqueName	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256
userPassword	primordial	String	Hashed Password	max-length=64

1.3.1.2 Relationship Paths for abstractIdentity (primordial)

The relationship paths shown in [Table 1–26](#), are defined in the primordial model.

Table 1–26 *abstractRole Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
originalGrantee	delegatedRoleGrants	one-to-many	roleGrant
grantee	role_grants	one-to-many	roleGrant
assignee	user_role_assignments	one-to-many	userRoleAssignment

1.3.1.3 Related audit objects for abstractIdentity

The related audit objects for `abstractIdentity`, shown in [Table 1–27](#), are defined in the standard model.

Table 1–27 *abstractIdentity Related Audit Objects*

Related Object Type	Incoming Relationship Path
roleGrant	grantee
roleGrant	originalGrantee
userRoleAssignment	assignee

1.3.2 person

The `person` object type extends `abstractIdentity` and represents users of Oracle Role Manager that are associated with a real person. Because of this association, they are often synchronized with external systems such as identity management, directory, or single-sign-on products. `person` objects are distinct from `systemIdentity` objects because they are typically synchronized with external systems and configured with custom attributes that would not be applicable for system identities.

The base definition of `person` is in the primordial model and it is extended in the standard model. `person` has the `manage`, `audit`, `grant`, and `delegate` permissions. The external title for `person` is "Person," which is used in tokenized system privilege names, for example "Audit Person objects."

1.3.2.1 person Attributes

The person object inherits attributes from `abstractIdentity`, defined in the primordial model, and also has direct attributes, defined in the standard model. All person attributes are shown in [Table 1–28](#).

Table 1–28 *person Attributes*

Inherited Attribute ID	Attribute ID	Defined in	Type	Default Title	Constraints
	<code>displayName</code>	primordial	String	Display Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	<code>locale</code>	primordial	String	Locale	max-length=128
	<code>status</code>	primordial	String	Status	Must not be null. Allowable values are <code>inactive</code> or <code>active</code> . Default is <code>inactive</code> . max-length=32
	<code>uniqueName</code>	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256
	<code>userPassword</code>	primordial	String	Hashed Password	max-length=64
	<code>audio</code>	standard	binary	Audio file	
	<code>businessCategory</code>	standard	String	Business Category	max-length=64
	<code>carLicense</code>	standard	String	Car License	max-length=16
	<code>departmentNumber</code>	standard	String	Department Number	max-length=64
	<code>description</code>	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	<code>destinationIndicator</code>	standard	String	Destination Indicator	max-length=256
	<code>employeeNumber</code>	standard	String	Employee Number	whitespace (cannot have leading or trailing spaces) max-length=32

Table 1–28 (Cont.) person Attributes

Inherited Attribute ID	Attribute ID	Defined in	Type	Default Title	Constraints
	employeeType	standard	String	Employee Type	max-length=64
	fax	standard	String	Fax number	max-length=32
	givenName	standard	String	First Name	whitespace (cannot have leading or trailing spaces) max-length=128
	homePhone	standard	String	Home phone number	max-length=32
	homePostalAddress	standard	String	Home Postal Address	max-length=256
	initials	standard	String	Initials	max-length=16
	jobTitle	standard	String	Job Title	whitespace (cannot have leading or trailing spaces) max-length=256
	jpegPhoto	standard	binary	JPEG Photo	
	l	standard	String	Locale	max-length=256
	mail	standard	String	Email	max-length=128
	mobile	standard	String	Mobile phone number	max-length=20
	pager	standard	String	Pager number	max-length=20
	photo	standard	binary	Photo	
	physicalDeliveryOfficeName	standard	String	Physical delivery office	max-length=256
	postalAddress	standard	String	Postal Address	max-length=256
	postalCode	standard	String	Postal Code	max-length=256
	postOfficeBox	standard	String	P.O. Box	max-length=256
	preferredDeliveryMethod	standard	String	Preferred Delivery Method	max-length=256
	preferredLanguage	standard	String	Preferred Language	max-length=256
	registeredAddress	standard	String	Registered Address	max-length=256
	roomNumber	standard	String	Room Number	max-length=256
	seeAlso	standard	String	See Also	max-length=256
	sn	standard	String	Last Name	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=256
	st	standard	String	State	max-length=256
	street	standard	String	Street	max-length=256

Table 1–28 (Cont.) person Attributes

Inherited Attribute ID	Attribute ID	Defined in	Type	Default Title	Constraints
	telephoneNumber	standard	String	Telephone Number	max-length=20
	telexNumber	standard	String	Telex Number	max-length=20
	userCertificate	standard	binary	User Certificate	
	userID	standard	string	User ID	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) min-length=4 max-length=256
	userSMIMECertificate	standard	binary	User SMIME Certificate	
	x121Address	standard	String	x121 Address	max-length=16

1.3.2.2 Relationship Paths for person

The relationship paths shown in [Table 1–29](#) are defined in the standard model.

Table 1–29 person Relationship Paths

Outgoing ID	Incoming ID	Cardinality	Foreign Type
headedOrg	orgHead	one-to-many	abstractOrg
secretary	secretarialCients	many-to-one	person
parent_reporting_organization	child_reporting_people	many-to-one	abstractOrg
parent_cost_center_organization	child_cost_center_people	many-to-one	abstractOrg
parent_location_organization	child_location_people	many-to-one	abstractOrg
manager	managedPeople	many-to-one	person
ownedITRoles	roleOwner	one-to-many	itRole
ownedSystemRoles	roleOwner	one-to-many	systemRole
ownedBusinessRoles	roleOwner	one-to-many	businessRole
ownedApproverRoles	roleOwner	one-to-many	approverRole

1.3.3 systemIdentity

The `systemIdentity` object type extends `abstractIdentity` and represents users of Oracle Role Manager that do not have a person representation. Examples of these are:

- System
 - This is the identity used by Oracle Role Manager to represent the user for automated tasks. This identity cannot be used externally as it has no password to authenticate with. This user is created through the bootstrap process and cannot be modified.
- System Administrator

This is the identity used to administer the system, for example, for running deployment tools at the command line or through the Oracle Universal Installer. This user is created through the bootstrap process and can be modified and deleted as needed.

- oimSystem

This is an identity used by an external system, Oracle Identity Manager, when it interacts with Oracle Role Manager for purposes such as system synchronization. The user is created as part of the deployment process of the Oracle Role Manager Integration Library. Refer to the *Oracle Role Manager Integration Guide* for information about deploying the Oracle Role Manager Integration Library and the oimSystem user. Refer to the *Oracle Role Manager Administrator's Guide* for information about creating system identities.

The definition of `systemIdentity` is in the primordial model. `systemIdentity` has the `manage`, `audit`, `grant`, and `delegate` permissions. The external title for `systemIdentity` is "System Identity," which is used in tokenized system privilege names, for example "Audit System Identity objects."

1.3.3.1 Attributes for systemIdentity

The attributes for `systemIdentity`, shown in [Table 1–30](#) are defined in the primordial model. All attributes are inherited from the `abstractIdentity` object type from which `systemIdentity` is extended.

When an external system is the system of record for people or any other particular object type, the Oracle Role Manager administrator would typically configure the system roles/grants so that only the system identity that represents the system of record for person would have permission to create, modify or delete person records. An example of this is the `oimSystem` system identity for Oracle Identity Manager.

Table 1–30 *systemIdentity Attributes*

Inherited Attribute ID	Attribute ID	Defined in	Type	Default Title	Constraints
displayName		primordial	String	Display Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
locale		primordial	String	Locale	max-length=128
status		primordial	String	Status	Must not be null. Allowable values are <code>inactive</code> or <code>active</code> . The default is <code>inactive</code> . max-length=32
uniqueName		primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256
userPassword		primordial	String	Hashed Password	max-length=64

Table 1–30 (Cont.) systemIdentity Attributes

Inherited Attribute ID	Attribute ID	Defined in	Type	Default Title	Constraints
	description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	userID	primordial	String	User ID	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) min-length=4 max-length=256

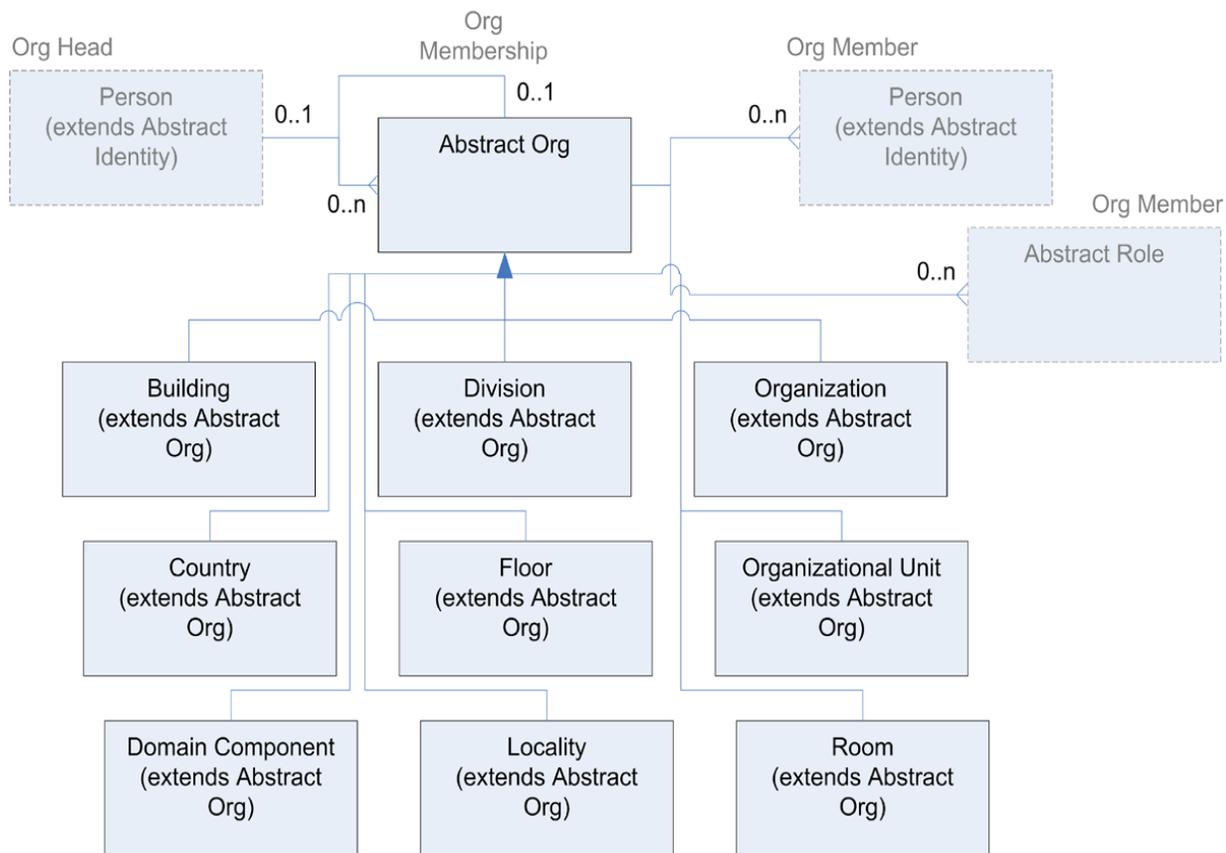
1.4 Organizations

The objects described in this section and illustrated in [Figure 1–3](#) represent various containers that are used to model different types of organization structures such as reporting hierarchies, locations and cost centers. These objects are defined in the standard model and not required for the Oracle Role Manager system to function. However, the Oracle Role Manager Web application requires that these object types exist.

The following objects are described in this section:

- [abstractOrg](#)
- [building](#)
- [country](#)
- [dcObject](#)
- [division](#)
- [floor](#)
- [locality](#)
- [organization](#)
- [ou](#)
- [room](#)

Figure 1–3 Organization Model Entity Relationship Diagram



1.4.1 abstractOrg

The `abstractOrg` object type represents the base type for all organizational constructs. This is necessary to allow generalization for things like hierarchy construction. Most interorganizational relationships are defined in `abstractOrg` so other objects generally refer to this base type for membership information.

An instance of `abstractOrg` is never explicitly created, but rather, is created as part of creating instances of its subtypes.

The definition of `abstractOrg` is in the standard model and has the `manage` and `audit` permissions. The external title for `abstractOrg` instances is "all organization," which is used in tokenized system privileges names, for example, "Audit all organization objects" and "Manage all organization objects."

1.4.1.1 Attributes for abstractOrg

The attributes for `abstractOrg`, shown in [Table 1–31](#) are defined in the standard model.

Table 1–31 abstractOrg Attributes

Attribute ID	Defined in	Type	Default Title	Constraints
description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000

Table 1–31 (Cont.) abstractOrg Attributes

Attribute ID	Defined in	Type	Default Title	Constraints
displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
uniqueName	primordial	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.4.1.2 Relationship Paths for abstractOrg (standard)

The relationship path shown in [Table 1–32](#), is defined in the standard model.

Of these relationship paths, all of the `parent_*_organization` paths have an incoming, restricted parent action, so that the `abstractOrg` cannot be deleted if there is an active relationship. An active relationship would be if an organization contains another organization, person, or role.

Note: This approach to generic organizational membership, allows the data model great flexibility but also could represent some nonintuitive patterns, such as `building` being contained in `floor`, and so forth.

Table 1–32 abstractOrg Relationship Paths (standard)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
orgHead	headedOrgs	many-to-one	person
parent_reporting_organization	child_reporting_organizations	many-to-one	abstractOrg
parent_location_organization	child_location_organizations	many-to-one	abstractOrg
parent_cost_center_organization	child_cost_center_organizations	many-to-one	abstractOrg
reporting_hierarchy_root	reporting_hierarchy_anchors	many-to-one	reportingHierarchy
location_hierarchy_root	location_hierarchy_anchors	many-to-one	locationHierarchy
cost_center_hierarchy_root	cost_center_hierarch_anchors	many-to-one	costCenterHierarchy
child_reporting_roles	parent_reporting_organization	one-to-many	abstractRole
child_location_roles	parent_location_organization	one-to-many	abstractRole
child_cost_center_roles	parent_cost_center_organization	one-to-many	abstractRole
child_reporting_people	parent_reporting_organization	one-to-many	person

Table 1–32 (Cont.) abstractOrg Relationship Paths (standard)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
child_location_people	parent_location_ organization	one-to-many	person
child_cost_center_people	parent_cost_center_ organization	one-to-many	person

1.4.2 building

The `building` subtype of `abstractOrg` is specifically for representing physical buildings. The definition of `building` is in the standard model and has the `manage` and `audit` permissions. The external title for `building` instances is "Building," which is used in tokenized system privilege names, for example "Manage Building objects."

The attributes for `building`, shown in [Table 1–33](#) are defined in the standard model. Many attributes are inherited from the `abstractOrg` object type from which `building` is extended.

Table 1–33 building attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
buildingName		standard	String	Building name	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=64
postalAddress		standard	String	Postal Address	max-length=256
telephoneNumber		standard	String	Telephone number	max-length=20
	description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=256

1.4.3 country

The `country` subtype of `abstractOrg` is specifically for representing countries. The definition of `country` is in the standard model and has the `manage` and `audit` permissions. The external title for `country` instances is "Country," which is used in tokenized system privilege names, for example "Manage Country objects."

The attributes for `country`, shown in [Table 1–34](#) are defined in the standard model. Many attributes are inherited from the `abstractOrg` object type from which `country` is extended.

Table 1–34 *country attributes*

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
c		standard	String	Country code	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=2
	description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=256

1.4.4 dcObject

The `dcObject` subtype of `abstractOrg` is specifically for representing domain components. The definition of `dcObject` is in the standard model and has the `manage` and `audit` permissions. The external title for `dcObject` instances is "Domain Component," which is used in tokenized system privilege names, for example "Manage Domain Component objects."

The attributes for `dcObject`, shown in [Table 1–35](#) are defined in the standard model. All attributes are inherited from the `abstractOrg` object type from which `dcObject` is extended.

Table 1–35 *dcObject attributes*

Inherited Attribute ID	Defined in	Type	Default Title	Constraints
description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256

Table 1–35 (Cont.) dcObject attributes

Inherited Attribute ID	Defined in	Type	Default Title	Constraints
uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=256

1.4.5 division

The `division` subtype of `abstractOrg` is specifically for representing divisions. The definition of `division` is in the standard model and has the `manage` and `audit` permissions. The external title for `division` instances is "Division," which is used in tokenized system privilege names, for example "Manage Division objects."

The attributes for `division`, shown in [Table 1–36](#) are defined in the standard model. All attributes are inherited from the `abstractOrg` object type from which `division` is extended.

Table 1–36 division attributes

Inherited Attribute ID	Defined in	Type	Default Title	Constraints
description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=256

1.4.6 floor

The `floor` subtype of `abstractOrg` is specifically for representing floors. The definition of `floor` is in the standard model and has the `manage` and `audit` permissions. The external title for `floor` instances is "Floor," which is used in tokenized system privilege names, for example "Manage Floor objects."

The attributes for `floor`, shown in [Table 1–37](#) are defined in the standard model. Most attributes are inherited from the `abstractOrg` object type from which `floor` is extended.

Table 1–37 *floor attributes*

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
floorIdentifier		standard	String	Floor identifier	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=64
	description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	standard		Display Name	whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.4.7 locality

The `locality` subtype of `abstractOrg` is specifically for representing groupings of locales. The definition of `locality` is in the standard model and has the `manage` and `audit` permissions. The external title for `locality` instances is "Locality," which is used in tokenized system privilege names, for example "Manage Locality objects."

The attributes for `locality`, shown in [Table 1–38](#) are defined in the standard model. Many attributes are inherited from the `abstractOrg` object type from which `locality` is extended.

Table 1–38 *locality attributes*

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
l		standard	String	Locale	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=256
seeAlso		standard	String	See Also	max-length=256
st		standard	String	State	max-length=256
street		standard	String	Street	max-length=256
	description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256

Table 1–38 (Cont.) locality attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
	uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.4.8 organization

The `organization` subtype of `abstractOrg` is specifically for representing organizations. The definition of `organization` is in the standard model and has the `manage` and `audit` permissions. The external title for `organization` instances is "Organization," which is used in tokenized system privilege names, for example "Manage Organization objects."

The attributes for `organization`, shown in [Table 1–39](#) are defined in the standard model. Some attributes are inherited from the `abstractOrg` object type from which `organization` is extended.

Table 1–39 organization attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
businessCategory		standard	String	Business category	max-length=64
destinationIndicator		standard	String	Destination Indicator	max-length=256
fax		standard	String	Fax number	max-length=32
internationalISDNNumber		standard	String	International ISDN Number	max-length=32
l		standard	String	Locale	max-length=256
physicalDeliveryOfficeName		standard	String	Physical delivery office	max-length=256
postalAddress		standard	String	Postal Address	max-length=256
postalCode		standard	String	Postal Code	max-length=256
postOfficeBox		standard	String	post Office Box	max-length=256
preferredDeliveryMethod		standard	String	Preferred Delivery method	max-length=256
registeredAddress		standard	String	Registered Address	max-length=256
seeAlso		standard	String	See Also	max-length=256
st		standard	String	State	max-length=256
street		standard	String	Street	max-length=256
telephoneNumber		standard	String	Telephone number	max-length=20
telexNumber		standard	String	Telex Number	max-length=20
x121Address		standard	String	x121Address	max-length=16

Table 1–39 (Cont.) organization attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
	description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=256

1.4.9 ou

The `ou` subtype of `abstractOrg` is specifically for representing organizational units. The definition of `ou` is in the standard model and has the `manage` and `audit` permissions. The external title for `ou` instances is "Organizational Unit," which is used in tokenized system privilege names, for example "Manage Organizational Unit objects."

The attributes for `ou`, shown in [Table 1–40](#) are defined in the standard model. Many attributes are inherited from the `abstractOrg` object type from which `ou` is extended.

Table 1–40 ou attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
businessCategory		standard	String	Business category	max-length=64
destinationIndicator		standard	String	Destination Indicator	max-length=64
fax		standard	String	Fax number	max-length=32
internationalISDNNumber		standard	String	International ISDN Number	max-length=32
l		standard	String	Locale	max-length=256
physicalDeliveryOfficeName		standard	String	Physical delivery office	max-length=256
postalAddress		standard	String	Postal Address	max-length=256
postalCode		standard	String	Postal Code	max-length=256
postOfficeBox		standard	String	post Office Box	max-length=256
preferredDeliveryMethod		standard	String	Preferred Delivery method	max-length=256
registeredAddress		standard	String	Registered Address	max-length=256
seeAlso		standard	String	See Also	max-length=256

Table 1–40 (Cont.) ou attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
st		standard	String	State	max-length=256
street		standard	String	Street	max-length=256
telephoneNumber		standard	String	Telephone number	max-length=20
telexNumber		standard	String	Telex Number	max-length=20
x121Address		standard	String	x121Address	max-length=16
	description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
	displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.4.10 room

The `room` subtype of `abstractOrg` is specifically for representing physical rooms. The definition of `room` is in the standard model and has the `manage` and `audit` permissions. The external title for `room` instances is "Room," which is used in tokenized system privilege names, for example "Manage Room objects."

The attributes for `room`, shown in [Table 1–41](#) are defined in the standard model. Some attributes are inherited from the `abstractOrg` object type from which `room` is extended.

Table 1–41 room attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
roomNumber		standard	String	Room number	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=256
seeAlso		standard	String	See Also	max-length=256
telephoneNumber		standard	String	Telephone number	max-length=20
	description	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000

Table 1–41 (Cont.) room attributes

Attribute ID	Inherited Attribute ID	Defined in	Type	Default Title	Constraints
	displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
	uniqueName	standard	String	Uniquely Identifying Name	Must be unique (case-sensitive=false) whitespace (cannot have leading or trailing spaces) max-length=256

1.5 Hierarchies

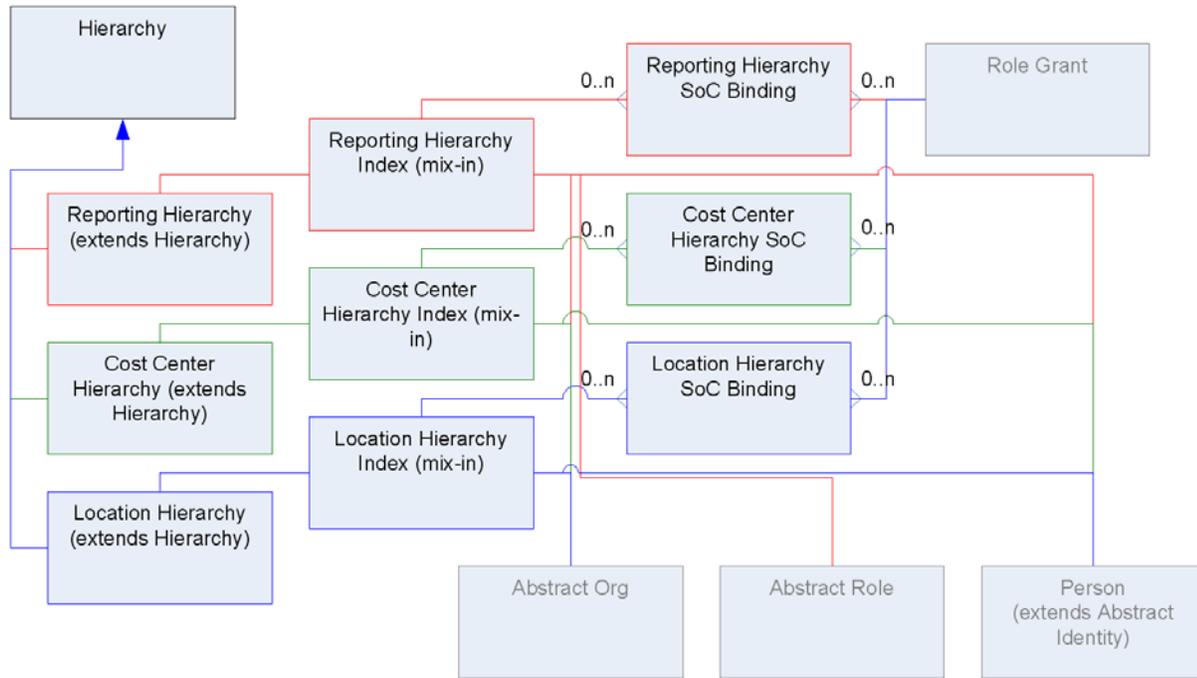
Hierarchies are a specialized combination between a type and a mechanism. Each defined hierarchy creates an index type to represent the hierarchy's index, and a *sphere of control* binding relationship that links role grants to the hierarchy index. The index type also acts as an attachment point for sphere of control. Sphere of control is a relationship between a grant and a node within a hierarchy such as reporting organization, cost center, and location. The sphere of control specifies the scope within which a grant is valid, which provides a means of limiting the validity of a role grant within a hierarchy.

The hierarchy model is illustrated in [Figure 1–4](#).

The following objects are described in this section:

- [hierarchy](#)
- [reportingHierarchy](#)
- [costCenterHierarchy](#)
- [locationHierarchy](#)

Figure 1–4 Hierarchy Model Entity Relationship Diagram



1.5.1 hierarchy

The `hierarchy` object is the abstract base type from which all hierarchy roots extend. It is defined in the primordial data model. It is necessary to support some generalizations, for example, the ability to specify hierarchical sphere of control for role grants.

The Oracle Role Manager deploy tool creates, edits and deletes `hierarchy` objects indirectly during the instance management of the hierarchy subtypes.

The definition of `hierarchy` is in the primordial model. The external title for `hierarchy` instances is "Hierarchy," which is used in system messages that display in the log files and application server console.

1.5.1.1 Attributes for hierarchy

The attributes for `hierarchy`, shown in [Table 1–42](#) are defined in the primordial model.

Table 1–42 abstractOrg Attributes

Attribute ID	Defined in	Type	Default Title	Constraints
displayName	standard	String	Display Name	Must not be null. whitespace (cannot have leading or trailing spaces) searchable=true max-length=256

1.5.1.2 Relationship Paths for hierarchy (primordial)

The relationship path shown in [Table 1–43](#), is defined in the primordial model.

Table 1–43 *hierarchy Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
socHierarchyRoles	socHierarchy	one-to-many	abstractRole

1.5.2 reportingHierarchy

The `reportingHierarchy` hierarchy object type represents the reporting structure of an organization. All object types use the reporting hierarchy to structure information.

A reporting hierarchy represents the managerial and authoritative relationships that reporting organizations have to one another. For example, the sales organization in China reports to the sales organization that spans all of Asia Pacific.

Because it is the default administrative hierarchy for the Oracle Role Manager Web application, the system places any unassigned person or role objects that do not have explicit parent organizations into a node in this hierarchy named Unassigned that is created at deploy time. (For more information about the Unassigned node, refer to the *Oracle Role Manager User's Guide*.)

The definition of `reportingHierarchy` is in the standard model and has the `manage` permission. The external title for `reportingHierarchy` instances is "Reporting Hierarchy Root," which is used in tokenized system privilege names, for example "Manage Reporting Hierarchy Root objects."

1.5.2.1 Node Relationships for reportingHierarchy

The node relationship paths shown in [Table 1–44](#), are defined in the standard model.

Table 1–44 *reportingHierarchy Node Relationship Paths (standard)*

Parent ID	Child ID	Relationship Path
reportingHierarchy	abstractOrg	reporting_hierarchy_root
abstractOrg	abstractOrg	parent_reporting_organization
abstractOrg	abstractRole	parent_reporting_organization
abstractOrg	person	parent_reporting_organization

1.5.3 costCenterHierarchy

The `costCenterHierarchy` hierarchy object type represents the cost center structure of the organization.

A cost center hierarchy represents the monetary relationships organizations have to one another. For example, the North American accounting organization pays the expenses for all organizations in North America.

The definition of `costCenterHierarchy` is in the standard model and has the `manage` permission. The external title for `reportingHierarchy` instances is "Cost Center Hierarchy Root," which is used in tokenized system privilege names, for example "Manage Cost Center Hierarchy Root objects."

1.5.3.1 Node Relationships for costCenterHierarchy

The node relationship paths shown in [Table 1–45](#), are defined in the standard model.

Table 1–45 *reportingHierarchy Node Relationship Paths (standard)*

Parent ID	Child ID	Relationship Path
costCenterHierarchy	abstractOrg	cost_center_hierarchy_root
abstractOrg	abstractOrg	cost_center_reporting_organization
abstractOrg	person	cost_center_reporting_organization

1.5.4 locationHierarchy

The `locationHierarchy` hierarchy object type represents the location structure of the organization.

The location hierarchy represents the physical relationship that different locations have to one another. For example, individual bank branches in San Francisco are organized under the California location.

The definition of `locationHierarchy` is in the standard model and has the `manage` permission. The external title for `locationHierarchy` instances is "Location Hierarchy Root," which is used in tokenized system privilege names, for example "Manage Location Hierarchy Root objects."

1.5.4.1 Node Relationships for locationHierarchy

The node relationship paths shown in [Table 1–46](#), are defined in the standard model.

Table 1–46 *locationHierarchy Node Relationship Paths (standard)*

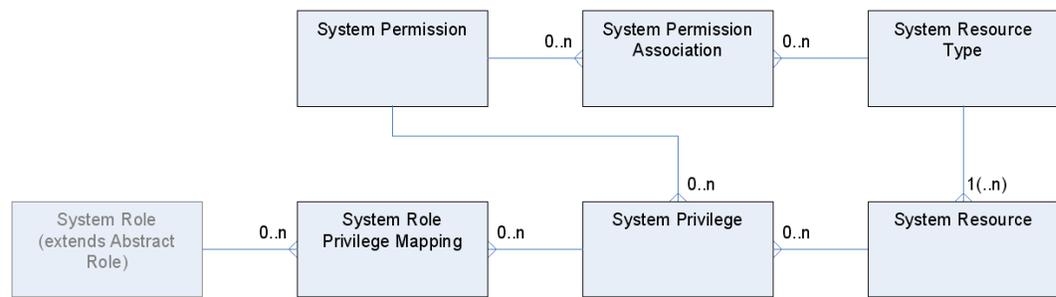
Parent ID	Child ID	Relationship Path
locationHierarchy	abstractOrg	location_hierarchy_root
abstractOrg	abstractOrg	parent_location_organization
abstractOrg	person	parent_location_organization

1.6 Internal Security and Access Control

The objects in this section are the underlying infrastructure used for the internal security model. (Refer to [Section 1.2.5](#) for details on the `systemRole` object type.)

The following objects are described in this section:

- [systemPermission](#)
- [systemPrivilege](#)
- [systemResource](#)
- [systemResourceType](#)
- [sysPermissionAssociation](#)
- [sysRolePrivilegeMapping](#)

Figure 1–5 Internal Security Entity Relationship Diagram

1.6.1 systemPermission

The `systemPermission` object type represents the permissions that are available to associate with system resource types to use for system privileges. They are explicitly defined in the data model configuration. The any permission is defined in the primordial model and implicitly inherits all of the capabilities of any other defined permission. At model deployment time, the deployment framework creates `systemPermission` objects based on the permissions defined in the data model configurations.

The definition of `systemPermission` is in the primordial model. The external title for `systemPermission` is "System Permission," which is used in which is used in system messages that display in the log files.

1.6.1.1 Attributes for systemPermission

The attributes for `systemPermission`, shown in [Table 1–54](#) are defined in the primordial model.

Table 1–47 systemPermission Attributes

Attribute ID	Defined in	Type	Default Title	Constraints
name	primordial	String	Name	Must not be null. Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256
displayName	primordial	String	Display Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000

1.6.1.2 Relationship Paths for systemPermission (primordial)

The relationship paths shown in [Table 1–48](#), are defined in the primordial model.

Table 1–48 *systemPermission Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
sys_applicable_permissions	system_permission	one-to-many	sysPermissionAssociation
n/a	system_permissions	many-to-many	systemResourceType
system_privileges	system_permission	one-to-many	systemPrivilege

1.6.2 systemPrivilege

The `systemPrivilege` object type represents the association between system permissions and system resources. `systemPrivilege` objects are mapped to system roles to represent that the members of that system role have the system permission with respect to the system resource (and due to the `all` pattern, to objects associated with that type). At model deployment time, the deployment framework creates one `systemPrivilege` for each `systemResource` for each applicable `systemPermission` (applicable system permissions are those mapped to the `systemResourceType` associated with the `systemResource`).

The base definition of `systemPrivilege` is in the primordial model and extended in the standard model. The external title for `systemPrivilege` is "System Privilege," which is used in which is used in system messages that display in the log files and application server console.

1.6.2.1 Attributes for systemPrivilege

The attributes for `systemPrivilege`, shown in [Table 1–49](#) are defined in the primordial model.

Table 1–49 *systemPrivilege Attributes*

Attribute ID	Defined in	Type	Default Title	Constraints
name	primordial	String	Name	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=256
displayName	primordial	String	Display Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000

1.6.2.2 Relationship Paths for systemPrivilege (primordial)

The relationship paths shown in [Table 1–50](#), are defined in the primordial model.

Table 1–50 *systemPrivilege Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
system_resource	system_privileges	many-to-one	systemResource
system_permission	system_privileges	many-to-one	systemPermission

Table 1–50 (Cont.) systemPrivilege Relationship Paths (primordial)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
sys_role_privilege_mappings	system_privilege	one-to-many	sysRolePrivilegeMapping

1.6.2.3 Related Audit Objects for systemPrivilege (primordial)

The related audit object, shown in [Table 1–51](#), is defined in the primordial model.

Table 1–51 systemPrivilege Related Audit Objects (primordial)

Related Object	Incoming Relationship Path
sysRolePrivilegeMapping	system_privilege

1.6.3 systemResource

The `systemResource` object type represents a particular instance or group of system resources. Each is associated with a `systemResourceType` and adds a qualifier. The qualifier is intended as a mechanism allowing system resources to represent some subset of all resources of that type. When the Oracle Role Manager model is first deployed, the deployment framework creates one `systemResource` instance with the `all` qualifier for each `systemResourceType` object. `systemResource` objects are created and updated by deployment framework and cannot be modified by users.

The definition of `systemResource` is in the primordial model. The external title for `systemResource` is "System Resource," which is used in system messages that display in the log files and application server console.

1.6.3.1 Attributes for systemResource

The attributes for `systemResource`, shown in [Table 1–52](#) are defined in the primordial model.

Table 1–52 systemResource Attributes

Attribute ID	Defined in	Type	Default Title	Constraints
name	primordial	String	Name	Must not be null. Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256
displayName	primordial	String	Display Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
qualifier	primordial	String	Qualifier	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=256

1.6.3.2 Relationship Paths for systemResource (primordial)

The relationship paths shown in [Table 1–53](#), are defined in the primordial model.

Table 1–53 *systemResource Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
system_resource_type	system_resources	many-to-one	systemResourceType
system_privileges	system_resource	one-to-many	systemPrivilege

1.6.4 systemResourceType

The `systemResourceType` object represents types of system resources (for example, `person`, `abstractRole`, `itRole`, or `organization`). When the Oracle Role Manager model is first deployed, the deployment framework creates one `systemResourceType` instance for each object type defined in the data model. `systemResourceType` objects cannot be modified by users and are normally not queried by them.

The definition of `systemResourceType` is in the primordial model. The external title for `systemResourceType` is "System Resource Type," which is used in system messages that display in the log files and application server console.

1.6.4.1 Attributes for systemResourceType

The attributes for `systemResourceType`, shown in [Table 1–54](#) are defined in the primordial model.

Table 1–54 *systemResourceType Attributes*

Attribute ID	Defined in	Type	Default Title	Constraints
name	primordial	String	Name	Must not be null. Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256
displayName	primordial	String	Display Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
description	primordial	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000

1.6.4.2 Relationship Paths for systemResourceType (primordial)

The relationship paths shown in [Table 1–55](#), are defined in the primordial model.

Table 1–55 *systemResourceType Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
system_resources	system_resource_type	one-to-many	systemResource
sys_applicable_permission	system_resource_type	one-to-many	sysPermissionAssociation
system_permissions	n/a	many-to-many	systemPermission

1.6.5 sysPermissionAssociation

The `sysPermissionAssociation` relationship object type represents the association between system permissions and a system resource types. If a system permission is associated to a system resource type, this signifies that the associated system permission is applicable for resources of that system resource type. These associations are defined at the object-type level in the data model configurations. There is an implicit association between all system resource types and the any primordial-defined system permission.

At model deployment time, the deployment framework creates these objects based upon the data model configuration.

The definition of `sysPermissionAssociation` is in the primordial model. The external title for `sysPermissionAssociation` is "System Resource Type to Permission Mapping," which is used in system messages that display in the log files and application server console.

1.6.5.1 Relationship Paths for relevantRoleAttribute (primordial)

The relationship path shown in [Table 1-56](#), is defined in the primordial model.

Table 1-56 *sysPermissionAssociation Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
system_resource_type	sys_applicable_permissions	many-to-one	systemResourceType
system_permission	sys_applicable_permissions	many-to-one	systemPermission

1.6.6 sysRolePrivilegeMapping

The `sysRolePrivilegeMapping` relationship object types are, along with `systemRole` objects, the only user-modifiable parts of the internal security model.

`sysRolePrivilegeMapping` objects describe which system privileges the members of a system role have. These mappings can be created or deleted through business logic to support whatever system role to system privilege combinations are required. At deployment time, the bootstrap deployment manager creates several of these to represent which privileges the system administrator role has by default.

The definition of `sysRolePrivilegeMapping` is in the primordial model. The external title for `sysRolePrivilegeMapping` is "System Role to System Privilege Mapping," which is used in system messages that display in the log files and application server console.

1.6.6.1 Relationship Paths for sysRolePrivilegeMapping (primordial)

The relationship path shown in [Table 1-57](#), is defined in the primordial model.

Table 1-57 *sysRolePrivilegeMapping Relationship Paths (primordial)*

Outgoing ID	Incoming ID	Cardinality	Foreign Type
system_role	sys_role_privilege_mappings	many-to-one	systemRole
system_privilege	sys_role_privilege_mappings	many-to-one	systemPrivilege

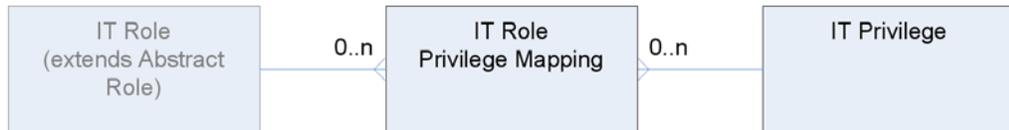
1.7 Entitlements

The objects described in this section represent external privilege information that may be consumed by external identity management and provisioning systems. These objects, illustrated in [Figure 1–6](#), have no impact on the internal security model.

The following objects are described in this section:

- [itPrivilege](#)
- [itRolePrivilegeMapping](#)

Figure 1–6 Entitlement Model Entity Relationship Diagram



1.7.1 itPrivilege

The `itPrivilege` object type represents an external entitlement. What that means to the external system is represented in the Oracle Role Manager system only in the `itPrivilegeDetails` CLOB attribute where information can be stored that is interpreted by the provisioning system or the actual system where the entitlement is applied. It is up to the external system, such as an access provisioning system, to interpret this data.

The base definition of `itPrivilege` is in the primordial model and extended in the standard model. `itPrivilege` has the `manage` and `audit` permissions. The external title for `itPrivilege` is "Entitlement," which is used in tokenized system privilege names, for example "Manage Entitlement objects."

1.7.1.1 Attributes for itPrivilege

The attributes for `itPrivilege`, shown in [Table 1–58](#), are defined in both the primordial model and the standard model.

Table 1–58 itPrivilege Attributes

Attribute ID	Defined in	Type	Default Title	Constraints
<code>description</code>	standard	String	Description	whitespace (cannot have leading or trailing spaces) max-length=4000
<code>displayName</code>	primordial	String	Entitlement Name	Must not be null whitespace (cannot have leading or trailing spaces) searchable=true max-length=256
<code>oimEntitlementId</code>	standard	integer	OIM Entitlement Identifier	Must be unique. scale=19
<code>itPrivilegeDetails</code>	primordial	CLOB	Entitlement Details	max-length=100000000

Table 1–58 (Cont.) itPrivilege Attributes

Attribute ID	Defined in	Type	Default Title	Constraints
resourceName	standard	String	Resource Name	max-length=256 whitespace (cannot have leading or trailing spaces)
uniqueName	standard	String	Uniquely Identifying name	Must be unique (case-sensitive=false). whitespace (cannot have leading or trailing spaces) max-length=256

1.7.1.2 Relationship Paths for itPrivilege (primordial)

The relationship path shown in [Table 1–59](#), is defined in the primordial model.

Table 1–59 itPrivilege Relationship Paths (primordial)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
it	system_privileges	many-to-one	systemResource

1.7.1.3 Related Audit Objects for itPrivilege (standard)

The related audit object, shown in [Table 1–60](#), is defined in the standard model.

Table 1–60 itPrivilege Related Audit Objects (standard)

Related Object	Incoming Relationship Path
itRolePrivilegeMapping	it_privilege

1.7.2 itRolePrivilegeMapping

The `itRolePrivilegeMapping` relationship object type represents the link between IT roles and entitlements. When a link between an IT role and an entitlement exists, this indicates that any member of mapped business role has the mapped entitlement.

The definition of `itRolePrivilegeMapping` is in the primordial model. The external title for `itRolePrivilegeMapping` is "IT Role to Entitlement Mapping," which is used in system messages that display in the log files and application server console.

1.7.2.1 Relationship Paths for itPrivilege (primordial)

The relationship path shown in [Table 1–59](#), is defined in the primordial model.

Table 1–61 itPrivilege Relationship Paths (primordial)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
it_role	it_role_privilege_mappings	many-to-one	itRole
it_privilege	it_role_privilege_mappings	many-to-one	itPrivilege

1.7.2.2 Related Audit Objects for itPrivilege (standard)

The related audit object, shown in [Table 1–60](#), is defined in the standard model.

Table 1–62 itPrivilege Related Audit Objects (standard)

Related Object	Incoming Relationship Path
itRolePrivilegeMapping	it_privilege

1.8 System Infrastructure

The objects in this section have little in common other than they are used by the system for (typically) non-business-related operations. Most of them are used for back-end infrastructure and have no obvious presence in any normal user interface. All of the objects described in this section are defined in the primordial model.

The following objects are described in this section:

- [auditEvent](#)
- [auditEventDetail](#)
- [baseBundle](#)
- [configuration](#)
- [localizedBundle](#)
- [pluginPack](#)

1.8.1 auditEvent

The `auditEvent` object type represents the audit events that are caused during the lifetime of business transactions. It is used by the business logic layer to search for audit events, but has special access patterns. The `auditEvent` objects that would be returned by a search are limited to those events where the current user has the appropriate audit system privilege over the event's related object. `auditEvent` objects are updated by the Oracle Role Manager framework during business transactions.

The definition of `auditEvent` is in the primordial model. The external title for `auditEvent` is "Audit Event," which is used in which is used in system messages that display in the log files and application server console.

1.8.1.1 Attributes for auditEvent

The attributes for `auditEvent`, shown in [Table 1–63](#), are defined in the primordial model.

Table 1–63 *auditEvent Attributes (primordial)*

Attribute ID	Defined in	Type	Default Title	Constraints
actorID	primordial	integer	Actor ID	Must not be null. scale=20
actorDisplayName	primordial	String	Actor Display Name	Must not be null. max-length=256
operationID	primordial	String	Operation ID	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=64
definitionID	primordial	String	Definition ID	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=64

Table 1–63 (Cont.) auditEvent Attributes (primordial)

Attribute ID	Defined in	Type	Default Title	Constraints
operationTitle	primordial	String	Operation Title	Must not be null. whitespace (cannot have leading or trailing spaces) max-length=128
reason	primordial	String	Reason	max-length=4000
eventTime	primordial	datetime	Event Time	Must not be null.
transactionID	primordial	integer	Transaction ID	Must not be null. scale=10
auditStatus	primordial	String	Audit Status	Must not be null. Allowable values are approved, canceled, finalized, pending, rejected, or submitted. max-length=32

1.8.1.2 Relationship Paths for auditEvent (primordial)

The relationship path shown in [Table 1–64](#) is defined in the primordial model.

Table 1–64 Relationship Paths for auditEvent

Outgoing ID	Incoming ID	Cardinality	Foreign Type
auditEvent	details	one-to-many	auditEventDetail

1.8.2 auditEventDetail

The `auditEventDetail` object type represents the audit event details that are created during the lifetime of business transactions. The `auditEventDetail` objects that would be returned by a search are limited to those events where the current user has the appropriate audit system privilege over the event's related object. `auditEventDetail` objects are updated by the Oracle Role Manager framework during business transactions.

The definition of `auditEventDetail` is in the primordial model. The external title for `auditEventDetail` is "Audit Event Detail," which is used in which is used in system messages that display in the log files and application server console.

1.8.2.1 Attributes for auditEventDetail

The attributes for `auditEventDetail`, shown in [Table 1–65](#), are defined in the primordial model.

Table 1–65 auditEventDetail Attributes (primordial)

Attribute ID	Defined in	Type	Default Title	Constraints
subjectKeyID	primordial	integer	subject Key ID	Must not be null. searchable=true scale=20

Table 1–65 (Cont.) auditEventDetail Attributes (primordial)

Attribute ID	Defined in	Type	Default Title	Constraints
subjectKeyType	primordial	String	Subject Key Type	Must not be null. searchable=true whitespace (cannot have leading or trailing spaces) max-length=64
changeTime	primordial	datetime	Change Time	Must not be null.
changeInfo	primordial	CLOB	Change Information XML	max-length=100000000
oldSubjectState	primordial	String	Old Subject State	Allowable values are effective or non-effective. max-length=64
newSubjectState	primordial	String	New Subject State	Allowable values are effective or non-effective. max-length=64

1.8.2.2 Relationship Paths for auditEventDetail (primordial)

The relationship path shown in [Table 1–66](#) is defined in the primordial model.

Table 1–66 Relationship Paths for auditEventDetail (primordial)

Outgoing ID	Incoming ID	Cardinality	Foreign Type
details	auditEvent	many-to-one	auditEvent

1.8.3 baseBundle

The `baseBundle` object type is where the default localization data is stored in XML form. It is used by the Oracle Role Manager deployment tools and not accessed for business purposes. The API base bundles are not in the database, but are instead in the class path.

1.8.4 configuration

The `configuration` object type is where the business configuration is stored in XML form. It is used by the Oracle Role Manager deployment tools and not accessed for business purposes.

1.8.5 localizedBundle

The `localizedBundle` object type is where the localizations are stored in property file form. It is used by the Oracle Role Manager deployment tools and read by way of the Oracle Role Manager administrative console; it is not accessed for business purposes. Both the `baseBundle` and the `localizedBundle` have a `categoryID` attribute indicating a relationship, but because some base bundles are deployed in the class path as opposed to the database, there cannot be relational integrity and hence no supporting relationship paths. This object is not accessed by business logic.

1.8.6 pluginPack

The `pluginPack` object type is where the pluggable custom Java business logic is stored in binary form. It is used by the Oracle Role Manager deployment tools and not accessed for business purposes.

Configuring the Data Model

This chapter describes the steps to modify the data model configuration of the Oracle Role Manager data model in a managed fashion.

Note: This chapter assumes that an instance of Oracle Role Manager is installed following the instructions in the *Oracle Role Manager Installation Guide*.

This chapter includes the following sections:

- [Best Practices](#)
- [Viewing the Standard Model Configuration](#)
- [Adding and Modifying Attributes](#)
- [Adding and Modifying Reference Attributes](#)
- [Adding Structural Type Objects](#)
- [Extending the Standard Configuration](#)
- [Modifying the Standard Configuration](#)
- [Deploying Data Model Customizations](#)

2.1 Best Practices

Caution: Data model changes cannot be reverted. Before making any modifications to the Oracle Role Manager data model, be sure to back up both of the Oracle Role Manager schemas (owner and application user schema) so that you can revert to the original state.

There are several approaches to customizing the Oracle Role Manager data model, depending on the type of customization that you want.

The data model is configurable through XML files. Oracle Role Manager provides a standard data model, which creates and configures the entities required by the Oracle Role Manager Web application. The first thing to determine is whether your customizations require modifications to the standard model or additions to the standard model.

If the standard model has been deployed and you need to make changes, it is recommended that you copy the standard configuration, edit it, then deploy your

changes to the database. If you are certain that your changes do not affect existing constraints, you can deploy your customizations over an existing deployment of the standard model.

Note: If you are using the Oracle Role Manager application's user interface, you should take care not to modify any entities or attributes in a way that would make the application unusable. Examples of this are removing the `organization` or `location` hierarchy entities.

If you are adding attributes on entities that are in the standard model or new entities with new definitions, you can create an XML file and deploy just those additions.

2.2 Viewing the Standard Model Configuration

The standard model depends on the *primordial* model, which defines the core entities required for Oracle Role Manager to function.

Both the standard and primordial models are described in detail in [Chapter 1](#). Although all the object types and constraints are described in that chapter, it is helpful to be aware of how those object types are defined in the XML file while determining the customization approach you want to take.

Note: The primordial model is not customizable.

To view the standard configuration, you must extract files from the `standard.car` configuration archive.

To view the standard model configuration:

1. On the Oracle Role Manager host, navigate to the `ORM_HOME/config` directory:
2. Using a utility like WinZip or `jar`, extract the entire contents of `standard.car` into a temporary location, such as `ORM_HOME/config_temp`.

Once expanded, notice that in your temporary location, there is the `config` directory, and in that, the subdirectories for the standard configuration. The subdirectory and files that pertain to the standard data model can be found in the following layout:

```
config/  
  oracle.iam.rm.temporal/  
    standard.xml  
    standard_permissions.xml
```

The `standard.xml` file contains configuration for all objects necessary for the Oracle Role Manager Web application to function properly.

2.3 Adding and Modifying Attributes

Attributes in the Oracle Role Manager model are first defined as *domains*. A domain is an abstract logical data type that when associated with an object type as part of the attribute definitions of an entity, it is then considered an *attribute*.

Adding attributes in Oracle Role Manager means the domain must be defined before it can be added to an attribute definition. In other words, the domain definition must

already exist, either in a configuration that your configuration depends on (such as standard), or earlier in the XML file in which it is being defined as an attribute.

Refer to the "Object Type Definition" and "Model Definition" sections of [Appendix A](#) for the schema definition defining which data types and constraints are available for use in attribute definitions.

Example 2–1 Creating an Attribute

This example adds an attribute to the person structural type with a pattern restriction for specified characters. (Refer to [Section 2.5](#) for more information about structural object types. (Refer to API documentation of the `java.util.regex.Pattern` class for information about controlling allowed characters in attribute values.)

This change is purely additive, so it can be deployed at any time after the standard model has been deployed.

In a new XML file, the new domain is first defined. For example:

```
<domain-definitions>
  <domain-definition name="favoriteColor">
    <t:string>
      <t:length id="length" max-length="32">
        <t:violation-message>The favorite color can be no longer than 32
characters</t:violation-message>
      </t:length>
    </t:string>
  </domain-definition>
</domain-definitions>
```

Later in the same file, the person structural type is extended to have the new domain as an attribute along with a special pattern to restrict specified characters. For example:

```
<type-definitions>
  <extension-type-definition extending-type-id="person">
    <attribute-definitions>
      <attribute domain="favoriteColor">
        <title>Favorite Color</title>
        <t:pattern id="pattern">
          <t:violation-message>The favorite color cannot contain the following
characters: # ^ + ! @ = </t:violation-message>
          <t:pattern>[^\#\^\+\!\@\=\s]+</t:pattern>
        </t:pattern>
      </attribute>
    </attribute-definitions>
  </extension-type-definition>
</type-definitions>
```

The model addition is then deployed as described in [Section 2.6, "Extending the Standard Configuration."](#)

Example 2–2 Modifying an Attribute

This example changes the `employeeNumber` attribute so that it is required for all person objects in the system. In addition, this example adds constraints so it is both required and unique (no duplicates may be created).

In the copy of the `standard.xml` file, in the section that extends the person object type, `<extension-type-definition extending-type-id="person">`, the

employeeNumber attribute is edited as follows (the modifications are shown in bold type):

```
<attribute domain="employeeNumber">
  <title>Employee Number</title>
  <t:string-ext>
    <t:non-null-constraint id="non-null">
      <t:violation-message>The employee number must be
specified.</t:violation-message>
    </t:non-null-constraint>
    <t:unique-constraint id="unique" case-sensitive="false"
type="varying-temporal">
      <t:violation-message>The employee number must be
unique.</t:violation-message>
    </t:unique-constraint>
    <t:whitespace id="whitespace">
      <t:violation-message>The employee number cannot have leading or trailing
spaces.</t:violation-message>
    </t:whitespace>
  </t:string-ext>
</attribute>
```

The model addition is then deployed as described in [Section 2.7, "Modifying the Standard Configuration."](#)

2.4 Adding and Modifying Reference Attributes

Relationships in Oracle Role Manager are defined by *reference attributes* in the object definition. Reference attributes are many-to-one relationships and refer to another object type rather than a primitive type.

Each reference attribute has an ID, the foreign object of the relationship and the IDs and user-friendly titles representing incoming and outgoing relationship paths.

For example, the person object definition contains this reference attribute for the relationship between person and costCenter as shown here:

```
<reference-attribute id="costCenterOrg_id" foreign-type="abstractOrg"
  parent-action="restrict"
  outgoing-relationship-path="parent_cost_center_organization"
  incoming-relationship-path="child_cost_center_people">
  <title>Cost Center Organization</title>
  <outgoing-relationship-path-title>Parent Cost Center
Organization</outgoing-relationship-path-title>
  <incoming-relationship-path-title>Child Cost Center
People</incoming-relationship-path-title>
</reference-attribute>
```

The following table describes the elements of this reference attribute. Refer to the [Appendix A.1, "Object Type Definition"](#) for the schema definition defining these elements.

Table 2–1 Reference Attribute Components

Attribute	Description
id	This is the ID of the reference attribute that backs this relationship path. The convention is that the ID be the object type followed by <code>_id</code> .

Table 2–1 (Cont.) Reference Attribute Components

Attribute	Description
foreign-type	The foreign type is the type of object this relationship points to. It can be the same object as the one used as the ID, or it can be its parent object. In the case of the relationship shown previously, the ID is <code>costCenterOrg_id</code> and the foreign type is <code>abstractOrg</code> . By defining the relationship using the parent type as the foreign type, this allows a broader relationship, for example, if the foreign type is <code>abstractOrg</code> , the parent can be any subtype of <code>abstractOrg</code> .
parent-action	This optional attribute is used for governing the sphere of control around relationships. The available values are: <ul style="list-style-type: none"> ▪ <code>restrict</code> The child must be deleted or the value changed before the parent can be deleted. ▪ <code>cascade</code> The child is deleted when the parent is deleted. ▪ <code>set-null</code> The child's field value in the relationship is set to null when the parent is deleted. <p>Note: If no parent action attribute is set, the default parent action is <code>set null</code>.</p>
outgoing-relationship-path	The outgoing relationship path represents the path from the object type where the reference attribute is being defined to the object type used as the ID.
incoming-relationship-path	The incoming relationship path represents the path from the object type used as the ID to the object type where the reference attribute is defined.

Example 2–3 Creating a Relationship

This example adds a reference attribute for the one-to-many, person to person relationship between a Review Manager and the people managed by that manager. This change is purely additive, therefore it can be deployed at any time after the standard model has been deployed.

In a new XML file, the `person` structural type is extended to have the new relationship. For example:

```
<type-definitions>
  <extension-type-definition extending-type-id="person">
    <attribute-definitions>
      <reference-attribute id="reviewManager_id" foreign-type="person"
        outgoing-relationship-path="reviewer_manager_person"
        incoming-relationship-path="reviewees_managed_people">
        <title>Review Manager</title>
        <outgoing-relationship-path-title>
          Review Manager
        </outgoing-relationship-path-title>
        <incoming-relationship-path-title>
          Review-Managed People
        </incoming-relationship-path-title>
      </reference-attribute>
    </attribute-definitions>
  </extension-type-definition>
</type-definitions>
```

The model addition is then deployed as described in [Section 2.6, "Extending the Standard Configuration."](#)

2.5 Adding Structural Type Objects

Structural type objects in the Oracle Role Manager model represent first-class business objects, such as organizations, people, and roles. They have no identifying foreign keys and store attributes and permissions.

Structural type objects can extend other structural type objects, for example like organization does from `abstractOrg`, or not, depending on the needs of the model.

When adding structural type objects, you can use existing domains in attribute definitions as well as any new domains that you need. Make sure that the domains are defined in the domain definitions before you associate them with the new object type.

Example 2-4 *Creating a Structural Type*

This example adds a new object type to represent a physical resource and its attributes. It also creates relationships (such the person responsible for the resource and the building that contains the resource) and permissions (manage and audit).

This change is purely additive, so it can be deployed at any time after the standard model has been deployed.

In a new XML file, all the new domains to associate with the new object are first defined. For example:

```
<domain-definitions>
  <domain-definition name="isInWorkingOrder">
    <t:boolean/>
  </domain-definition>
  <domain-definition name="isReservable">
    <t:boolean/>
  </domain-definition>
</domain-definitions>
```

Later in the same file, the `physicalResource` structural type is defined and the previously defined attributes and relationships are associated. For example:

```
<type-definitions>
  <structural-type-definition id="physicalResource">
    <title>Physical Resource</title>
    <attribute-definitions>
      <attribute domain="displayName">
        <title>Physical Resource</title>
        <t:string-ext>
          <t:non-null-constraint id="non-null">
            <t:violation-message>The resource name is
required.</t:violation-message>
          </t:non-null-constraint>
        </t:string-ext>
      </attribute>
      <attribute domain="description">
        <title>Description</title>
      </attribute>
      <attribute domain="isInWorkingOrder">
        <title>In Working Order</title>
        <t:boolean-ext>
          <t:default-value>true</t:default-value>
        </t:boolean-ext>
      </attribute>
    </attribute-definitions>
  </structural-type-definition>
</type-definitions>
```

```

        </t:boolean-ext>
    </attribute>
    <reference-attribute id="resourceOwner_id" foreign-type="person"
        outgoing-relationship-path="resourceOwner"
        incoming-relationship-path="ownedResources">
        <title>Resource Owner</title>
        <outgoing-relationship-path-title>Resource
    Owner</outgoing-relationship-path-title>
        <incoming-relationship-path-title>Owned
    Resources</incoming-relationship-path-title>
    </reference-attribute>
    <reference-attribute id="building_id" foreign-type="abstractOrg"
        outgoing-relationship-path="parent_building"
        incoming-relationship-path="child_resources">
        <title>Building</title>
        <outgoing-relationship-path-title>Parent Resource
    Building</outgoing-relationship-path-title>
        <incoming-relationship-path-title>Child Building
    Resources</incoming-relationship-path-title>
    </reference-attribute>
</attribute-definitions>
<permissions>
    <permission id="audit"/>
    <permission id="manage"/>
</permissions>
<access-policy read-audit-details-permission="audit"/>
</structural-type-definition>
</type-definitions>
    
```

The model addition is then deployed as described in [Section 2.6](#).

2.6 Extending the Standard Configuration

To extend the standard model, you must create an XML file containing your customizations and create a CAR file containing that XML file in the expected directory layout. This file depends on the standard model also being deployed.

To extend the standard model configuration:

1. On the Oracle Role Manager host, create a directory for your customization. For example, `C:\ORM_HOME\model_custom`.
2. In the new directory, create a subdirectory named `config`.
3. In the `config` directory, create a subdirectory named `oracle.iam.rm.temporal`.
4. In the `temporal` directory, create an XML file that contains the following expected elements at the beginning:

```

<?xml version="1.0" encoding="UTF-8"?>
<temporal xmlns="http://xmlns.oracle.com/iam/rm/temporal/config/1_0"
    xmlns:t="http://xmlns.oracle.com/iam/rm/type/def/1_0"
    id="datamodel_custom" version="1.0.0">

    <dependencies>
        <dependency id="standard" version="3.0.2"/>
    </dependencies>
    
```

where `datamodel_custom` is the name of the XML file without the file extension and `1.0.0` is the version to assign to this customization.

Note: For ongoing or incremental configuration changes, you must increment the version number and include the previously deployed version. For example, to increment from version 1.0.0 of a custom configuration, include the following in the `temporal` element:

```
version="1.0.1" previous-version="1.0.0"
```

5. Add the configuration you want in the body of the XML, after the `dependencies` element, in the following order:
 - New domain definitions
 - New permission definitions
 - New or extensions to object type definitions
6. Make sure the content in this file ends with:

```
</temporal>
```

7. Using a utility like WinZip or `jar`, repack everything in the within the directory created in step 1, and create an archive file appended with the `.car` file extension, for example, `datamodel_custom.car`.
8. Open the CAR file and ensure that it contains the XML file with the path `config/oracle.iam.rm.temporal`.

Note: If it instead is `datamodel_custom/config/oracle.iam.rm.temporal`, then your changes will not be deployed to the database.

9. Deploy your model additions by include this file in the collection of CAR files following the steps in [Section 2.8, "Deploying Data Model Customizations."](#)

2.7 Modifying the Standard Configuration

To modify the standard model, you must edit and repack the standard configuration before deploying your customizations to the database.

Note: Care should be taken when modifying object types in the standard model. If you are using the Oracle Role Manager Web application, changes you make to existing object types in this file could result in errors or unexpected behavior in the application.

To modify the standard model configuration:

1. From the temporary location where `standard.car` was extracted (refer to [Section 2.2](#)), navigate to `config/oracle.iam.rm.temporal`.
2. Copy the `standard.xml` file to create a duplicate file and give it a unique name, such as `standard_custom.xml`.
3. Add or edit the definitions in the copied file as needed.

Note: If you are deploying a model that has been deployed previously, you must increment the version number.

4. Using a utility like WinZip or jar, repackage everything in the temporary directory and create a file appended with the .car file extension, for example, `datamodel_custom.car`.
5. Include this file in the collection of CAR files as part of the deploy command described in [Section 2.8](#).

2.8 Deploying Data Model Customizations

This procedure assumes that Oracle Role Manager has been successfully installed and the Oracle Role Manager database instance is running.

To deploy data model customizations:

1. On the Oracle Role Manager installation host, navigate to `ORM_HOME/config`.
2. Make sure the `db.properties` file in `ORM_HOME/config` contains the correct information for your database environment. If it does not, modify it so it contains the following two lines:

```
db.driverClass=oracle.jdbc.driver.OracleDriver
db.connection_string=jdbc:oracle:thin:@//$HOST$: $PORT$/ $SERVICE$
```

where `$HOST$` is the database host name, `$PORT$` is the database listener port, and `$SERVICE$` is the database instance on which the Oracle Role Manager users/schemas were created.

3. Stop the Oracle Role Manager application server if it is running.
4. In a command window, navigate to `ORM_HOME/bin`.
5. Run the following command to deploy the configuration

```
deploy "collection_of_cars" orm-owner ormapp-user admin-user
```

where:

`collection_of_cars` contains the relative paths and file names of the CAR files to deploy. This collection must be within quotes with delimiters appropriate to the platform (a semicolon (;) for Windows, otherwise a colon (:)). For example, if you have customized the standard model and a separate customization that extends the standard model, the .car collection might be:

```
"../model_custom/datamodel_custom.car;../model_custom/standard_custom.car;"
```

`orm-owner` is the user name of the Oracle Role Manager database owner user/schema

`ormapp-user` is the user name of the Oracle Role Manager application user/schema

`admin-user` is the user name of the Oracle Role Manager System Administrator.

6. At the prompts, type the passwords of the database owner, application user, and administrator.

After you have deployed your data model customizations, you can verify the data model changes using a database utility such as Oracle SQL Developer or DB Visualizer.

Configuring the User Interface

This chapter describes the steps to modify the application user interface of Oracle Role Manager in a managed fashion.

This chapter includes the following sections:

- [Best Practices](#)
- [Extracting Files from the Web Application Archive](#)
- [Modifying Appearance and Style](#)
- [Modifying the Search Component](#)
- [Deploying UI Customizations](#)

3.1 Best Practices

Changes to the Oracle Role Manager user interface are made by modifying files contained in the Web application archive file that is deployed on the application server. To support future upgrades and to simplify reverting to the original state of the application, the recommendation is as follows:

- Archive a copy of the original Web application archive file (webui.ear or webui.war, depending on the application server) provided with the release.

Keeping a copy allows the opportunity to make comparisons and redeploy the application as its original state at any time.

- Extract the entire contents of the Web application archive file, retaining all directory path information, and store them in a version control repository system.

Using a version control system enables you to carefully manage updates and additions to those files, but more importantly, helps you retain your customizations when upgrading to later versions of Oracle Role Manager.

Note: All user interface customizations require modifying the content of the Web application archive file, rebundling it, and redeploying it on the application server. For most deployments, the application server must then be restarted for changes to be in effect.

For the purpose of clarity in this document, the location where you extracted the contents of the Web application archive file is referred to as *ORMUI_HOME*.

3.2 Extracting Files from the Web Application Archive

To extract files from Web application archive:

1. On the Oracle Role Manager host, navigate to the *ORM_HOME/webui* directory.
2. Go to the subdirectory named for the application server where Oracle Role Manager intended to be deployed (for example, *weblogic/10.3*).
3. In the source control location that you will store your customization, create a directory named *webui*.
4. Using a utility like WinZip or *jar*, extract the entire contents of the *webui.war* file (or for WebSphere, the *webui.ear* file) into the newly created *webui* directory.

Note: This location is the one referred to in this document as *ORMUI_HOME*.

Once expanded, the subdirectories and files you should see in that location are:

```
images/  
META-INF/  
nlstree/  
pages/  
scripts/  
styles/  
WEB-INF/  
signin.jsf  
signin.xhtml  
signout.xhtml
```

5. Navigate to *WEB-INF/lib* and create a directory named *webui*.
6. Using a utility like WinZip or *jar*, extract the entire contents of the *webui.jar* file into the newly created *webui* directory within *WEB-INF/lib*.

These files include the *Resource.properties* file that is used for setting properties and labels in the user interface.

7. Using the source control system, check in all these files so you can store the baseline of the application before any changes are made

3.3 Modifying Appearance and Style

The appearance and style of the Oracle Role Manager user interface is governed by style sheets and image files. The style sheet for the general areas of the application is the *style.css* file. This is where you can modify the "look and feel" of the application.

Before you decide on your modifications you might want to see the settings in this style sheet.

To view the style sheet:

- Navigate to the *ORMUI_HOME/styles* directory.

You should see the *style.css* file, which is where customizations can be made.

Once you decide on your modifications, you can edit this file and then deploy your changes as described in [Section 3.5](#).

The following sections include examples of some of the style modifications you can do.

3.3.1 Changing the Header Logo

If desired, you can replace the Oracle Role Manager logo with one of your own.

Note: The logo you use should be within the range of 234 pixels in width and 40 pixels in height.

Example 3-1 Changing the Logo

This example replaces the standard logo with a custom logo.

1. Navigate to `ORMUI_HOME/images`.
2. Rename `logo.gif` to `RoleManagerLogo.gif`.
3. Copy the logo you want to use as the banner logo into `ORMUI_HOME/images`.
4. Rename your logo to `logo.gif`.

Depending on your logo, you may want to modify the header background and link colors as described in the subsequent section.

3.3.2 Changing the Header Background and Link Colors

If desired, you can modify the header background and link colors in the header to match your logo.

Example 3-2 Changing the Background and Link Colors

This example changes the background of the header area to white. It also changes the color of the link text so they display over the white background.

1. Navigate to `ORMUI_HOME/styles`.
2. Open the `style.css` file for editing.
3. Change the `globalheader` element to remove the header background graphic and change the background color as follows:

```
.globalheader {
    background-color: #ffffff;
    padding-top: 10px;
    padding-bottom: 10px;
    width: 100%;
}
```

4. Change the `globalhelp` element to change the help links in the header to blue so they are visible on the white background as follows:

```
.globalhelp {
    font-size: 11px;
    font-family: Tahoma, Arial, sans-serif;
    color: #0033cc;
    vertical-align: top;
    text-align: right;
    padding-top: 5px;
    padding-right: 12px;
}
```

5. Change the following elements to change the color to black to make the Sign Out link visible:

```
..globalhelp a{
  color: #000000;
  text-decoration: none;
  text-align: right;
  padding-left: 5px;
}

.globalhelp a:visited{
  color: #000000;
  text-decoration: none;
  text-align: right;
}

.globalhelp a:hover {
  color: #000000;
  text-decoration: underline;
  text-align: right;
}
```

6. Save and close the style.css file.

3.4 Modifying the Search Component

The number of search results per page in can be configured by modifying the value for the `page.table.rowcount` property.

Example 3–3 *Changing the Number of Rows in the Search Results*

This example changes the number of rows in the search results area from the default of 15 rows to 20 rows.

1. Navigate to `ORMUI_HOME/WEB-INF/lib/webui`.
2. Open the `Resource.properties` file for editing.
3. Search for the property named `page.table.rowcount` and replace the value with 20 for twenty rows in the search results.
4. Save and close the `Resource.properties` file.
5. Using a utility such as WinZip or jar, rebundle the entire contents of `WEB-INF/lib/webui` and name the file `webui.jar` to replace the previous `webui.jar` file in `WEB-INF/lib`.
6. Remove the `webui` directory created in `ORMUI_HOME/WEB-INF/lib`.
7. Deploy your changes as described in [Section 3.5](#).

3.5 Deploying UI Customizations

This procedure assumes that Oracle Role Manager has been deployed previously following the instructions in the *Oracle Role Manager Installation Guide*.

To deploy UI customizations:

1. Navigate to the `ORMUI_HOME` directory chosen in [Section 3.2](#).
2. Using a utility such as WinZip or jar, archive the entire content of the `ORMUI_HOME` and save it with the following name:

- For WebLogic: `webui.war`
 - For JBoss: `webui.war`
 - For WebSphere: `webui.ear`
3. For WebLogic, deploy and test the modified Web application as follows:
 - a. In a Web browser, log in to the WebLogic Server Console. For example:
`http://appserverhost:7001/console`
 - b. In the left pane, click **Deployments**, then click **Install**.
 - c. Navigate to the location of the new Web application archive file, select `webui.war`, then click **Next**.
 - d. Select **Install this deployment as an application**, then click **Next**.
 - e. In the **Target** field, select the Oracle Role Manager server, then click **Next**.
 - f. In the **Name** field, enter `Oracle Role Manager Application`.
 - g. In the Security list, select **Custom Roles and Policies: Use only roles and policies that are defined in the Administration Console**, then click **Next**.
 - h. Click **Finish**.
 - i. In a Web browser, navigate to the Oracle Role Manager Web application address. For example:
`http://appserverhost:7001/webui`
 - j. Log in as the Oracle Role Manager Administrator.
You should be able to see your customizations to the UI.
 4. For JBoss, deploy and test the modified Web application as follows:
 - a. Copy the new `webui.war` file to the JBoss server where Oracle Role Manager server is deployed. For example:
`JBOSS_HOME/server/default/deploy`
 - b. Start the JBoss server.
 - c. In a Web browser, navigate to the Oracle Role Manager Web UI. For example:
`http://appserverhost:8080/webui`
 - d. Log in as the Oracle Role Manager Administrator.
You should be able to see your customizations to the UI.
 5. For WebSphere, deploy and test the modified Web application as follows:
 - a. In a Web browser, log in to the WebSphere administrative console. For example:
`http://appserverhost:9060/ibm/console`
 - b. From **Applications > Install New Application**, choose **Remote file system**, then click **Browse** to navigate to the location of the new Web application archive file, then select `webui.ear`.
 - c. Click **Next** on the next two pages to accept the defaults.

- d. Click **Finish**, then save your changes.
- e. From **Applications > Enterprise Applications > ORM Web UI**, click **Manage Modules**.
- f. Select **webui**.
- g. In the Class loader order list, choose **Classes loaded with application class loader first**.
- h. Click **OK**, then save your changes.
- i. From Applications > Enterprise Applications, select **ORM Web UI**, then click **Start**.

(This assumes you are administering WebSphere on the same server as the ORM Web UI is installed.)

- j. In a Web browser, navigate to the Oracle Role Manager Web application address. For example:

`http://appserverhost:9080/webui`

- k. Log in as the Oracle Role Manager Administrator.
You should be able to see your customizations to the UI.

XML Schema Definitions

This appendix contains the XML schema definitions for the data model. The standard data model and any customizations to the Oracle Role Manager model are validated by two schema definitions, the object type definition and the model configuration.

This appendix contains the following sections:

- [Object Type Definition](#)
- [Model Definition](#)

A.1 Object Type Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://xmlns.oracle.com/iam/rm/type/def/1_0"
  targetNamespace="http://xmlns.oracle.com/iam/rm/type/def/1_0"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified" >

  <xsd:simpleType name="Version">
    <xsd:restriction base="xsd:string">
      <xsd:pattern
        value="[0-9]{1,4}\.[0-9]{1,4}\.[0-9]{1,4}(\.[0-9]{1,4}){0,2}">
        <xsd:annotation>
          <xsd:documentation>i.e. x.y.z, x.y.z.j or x.y.z.j.h. where
            x, y, z, j and h are "small" integers</xsd:documentation>
        </xsd:annotation>
      </xsd:pattern>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="StringType">
    <xsd:complexContent>
      <xsd:extension base="PrimitiveType">
        <xsd:sequence>
          <xsd:element name="length" type="StringLengthConstraint"
            minOccurs="0"/>
          <xsd:element name="length" type="StringLengthConstraint"/>
          <xsd:element name="whitespace" type="WhitespaceConstraint"
            minOccurs="0"/>
          <xsd:element name="pattern" type="StringPatternConstraint"
            minOccurs="0"/>
          <xsd:element name="default-value" type="xsd:string"
            minOccurs="0"/>
          <xsd:element name="default-unique-value" type="xsd:string"
```

```
        minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="StringTypeExt">
    <xsd:complexContent>
        <xsd:extension base="PrimitiveTypeExt">
            <xsd:sequence>
                <xsd:element name="whitespace" type="WhitespaceConstraint"
                    minOccurs="0"/>
                <xsd:element name="pattern" type="StringPatternConstraint"
                    minOccurs="0"/>
                <xsd:element name="default-value" type="xsd:string"
                    minOccurs="0"/>
                <xsd:element name="default-unique-value" type="xsd:string"
                    minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType abstract="true" name="NumericType">
    <xsd:complexContent>
        <xsd:extension base="PrimitiveType">
            <xsd:sequence>
                <xsd:element name="value-range" type="NumericConstraint"
                    minOccurs="0"/>
            </xsd:sequence>
            <xsd:attribute name="scale" type="xsd:nonNegativeInteger"
                use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType abstract="true" name="NumericTypeExt">
    <xsd:complexContent>
        <xsd:extension base="PrimitiveTypeExt">
            <xsd:sequence>
                <xsd:element name="value-range" type="NumericConstraint"
                    minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="IntegerType">
    <xsd:complexContent>
        <xsd:extension base="NumericType">
            <xsd:sequence>
                <xsd:element name="default-value" type="xsd:long"
                    minOccurs="0"/>
                <xsd:element name="default-unique-value" type="xsd:string"
                    minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="IntegerTypeExt">
  <xsd:complexContent>
    <xsd:extension base="NumericTypeExt">
      <xsd:sequence>
        <xsd:element name="default-value" type="xsd:long"
          minOccurs="0"/>
        <xsd:element name="default-unique-value" type="xsd:string"
          minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DecimalType">
  <xsd:complexContent>
    <xsd:extension base="NumericType">
      <xsd:sequence>
        <xsd:element name="default-value" type="xsd:double"
          minOccurs="0"/>
        <xsd:element name="default-unique-value" type="xsd:string"
          minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="precision" type="xsd:nonNegativeInteger"
        use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DecimalTypeExt">
  <xsd:complexContent>
    <xsd:extension base="NumericTypeExt">
      <xsd:sequence>
        <xsd:element name="default-value" type="xsd:double"
          minOccurs="0"/>
        <xsd:element name="default-unique-value" type="xsd:string"
          minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="DateTimeType">
  <xsd:complexContent>
    <xsd:extension base="PrimitiveType">
      <xsd:sequence>
        <xsd:element name="default-value" type="dateTime"
          minOccurs="0"/>
        <xsd:element name="default-unique-value" type="xsd:string"
          minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:simpleType name="dateTime">
  <xsd:restriction base="xsd:string">
    <!-- yyyy-MM-dd'T'HH:mm:ss.SSS OR "transaction" -->
    <xsd:pattern
      value="(\\d{4}-(0[1-9]|1[0-2])-(0[1-9]|[12]\\d|3[01])T([01]\\d|2[
        0-3])|(:[0-5]\\d){2}(\\.\\d{3})?)|transaction"/>
    </xsd:restriction>
  </xsd:simpleType>

<xsd:complexType name="DateTimeTypeExt">
  <xsd:complexContent>
    <xsd:extension base="PrimitiveTypeExt">
      <xsd:sequence>
        <xsd:element name="default-value" type="dateTime"
          minOccurs="0"/>
        <xsd:element name="default-unique-value" type="xsd:string"
          minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="BooleanType">
  <xsd:complexContent>
    <xsd:extension base="PrimitiveType">
      <xsd:sequence>
        <xsd:element name="default-value" type="xsd:boolean"
          minOccurs="0"/>
        <xsd:element name="default-unique-value" type="xsd:string"
          minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="BooleanTypeExt">
  <xsd:complexContent>
    <xsd:extension base="PrimitiveTypeExt">
      <xsd:sequence>
        <xsd:element name="default-value" type="xsd:boolean"
          minOccurs="0"/>
        <xsd:element name="default-unique-value" type="xsd:string"
          minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="BinaryType">
  <xsd:complexContent>
    <xsd:extension base="AbstractType"/>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="BinaryTypeExt">
  <xsd:complexContent>
    <xsd:extension base="PrimitiveTypeExt"/>
  </xsd:complexContent>
</xsd:complexType>
```

```

<xsd:complexType abstract="true" name="PrimitiveType">
  <xsd:complexContent>
    <xsd:extension base="AbstractType" />
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType abstract="true" name="PrimitiveTypeExt">
  <xsd:complexContent>
    <xsd:extension base="AbstractTypeExt" />
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ReferenceType">
  <xsd:complexContent>
    <xsd:extension base="AbstractType">
      <xsd:attribute name="object-type" type="xsd:string"
        use="required" />
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ReferenceTypeExt">
  <xsd:complexContent>
    <xsd:extension base="AbstractTypeExt" />
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType abstract="true" name="AbstractType">
  <xsd:sequence>
    <xsd:element name="non-null-constraint" type="NonNullConstraint"
      minOccurs="0" />
    <xsd:element name="unique-constraint" type="UniqueConstraint"
      minOccurs="0" />
    <xsd:element name="multi-value-range-constraint"
      type="MultiValueRangeConstraint" minOccurs="0" />
    <xsd:element name="valid-value-constraint" type="ValidValueConstraint"
      minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="multi-value" type="xsd:boolean" use="optional"
    default="false" />
</xsd:complexType>

<xsd:complexType abstract="true" name="AbstractTypeExt">
  <xsd:sequence>
    <xsd:element name="non-null-constraint" type="NonNullConstraint"
      minOccurs="0" />
    <xsd:element name="unique-constraint" type="UniqueConstraint"
      minOccurs="0" />
    <xsd:element name="multi-value-range-constraint"
      type="MultiValueRangeConstraint" minOccurs="0" />
    <xsd:element name="valid-value-constraint" type="ValidValueConstraint"
      minOccurs="0" />
  </xsd:sequence>
  <xsd:attribute name="multi-value" type="xsd:boolean" use="optional"
    default="false" />
</xsd:complexType>

```

```
<xsd:group name="TypeChoice">
  <xsd:choice>
    <xsd:element name="string" type="StringType"/>
    <xsd:element name="boolean" type="BooleanType"/>
    <xsd:element name="integer" type="IntegerType"/>
    <xsd:element name="decimal" type="DecimalType"/>
    <xsd:element name="datetime" type="DateTimeType"/>
    <xsd:element name="reference" type="ReferenceType"/>
    <xsd:element name="binary" type="BinaryType"/>
  </xsd:choice>
</xsd:group>

<xsd:group name="TypeExtChoice">
  <xsd:choice>
    <xsd:element name="string-ext" type="StringTypeExt"/>
    <xsd:element name="boolean-ext" type="BooleanTypeExt"/>
    <xsd:element name="integer-ext" type="IntegerTypeExt"/>
    <xsd:element name="decimal-ext" type="DecimalTypeExt"/>
    <xsd:element name="datetime-ext" type="DateTimeTypeExt"/>
    <xsd:element name="reference-ext" type="ReferenceTypeExt"/>
    <xsd:element name="binary-ext" type="BinaryTypeExt"/>
  </xsd:choice>
</xsd:group>

<xsd:group name="PrimitiveTypeExtChoice">
  <xsd:choice>
    <xsd:element name="string-ext" type="StringTypeExt"/>
    <xsd:element name="boolean-ext" type="BooleanTypeExt"/>
    <xsd:element name="integer-ext" type="IntegerTypeExt"/>
    <xsd:element name="decimal-ext" type="DecimalTypeExt"/>
    <xsd:element name="datetime-ext" type="DateTimeTypeExt"/>
    <xsd:element name="binary-ext" type="BinaryTypeExt"/>
  </xsd:choice>
</xsd:group>

<xsd:group name="PrimitiveTypeChoice">
  <xsd:choice>
    <xsd:element name="string" type="StringType"/>
    <xsd:element name="boolean" type="BooleanType"/>
    <xsd:element name="integer" type="IntegerType"/>
    <xsd:element name="decimal" type="DecimalType"/>
    <xsd:element name="datetime" type="DateTimeType"/>
    <xsd:element name="binary" type="BinaryType"/>
  </xsd:choice>
</xsd:group>

<xsd:complexType abstract="true" name="ValueConstraint">
  <xsd:sequence>
    <xsd:element name="violation-message" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
</xsd:complexType>
```

```
<xsd:complexType name="NumericConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:attribute name="min-value" type="xsd:double" use="optional"/>
      <xsd:attribute name="max-value" type="xsd:double" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="StringLengthConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:attribute name="min-length" type="xsd:nonNegativeInteger"
        use="optional" default="0"/>
      <xsd:attribute name="max-length" type="xsd:nonNegativeInteger"
        use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="WhitespaceConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:attribute name="allow-leading-spaces" type="xsd:boolean"
        use="optional" default="false"/>
      <xsd:attribute name="allow-trailing-spaces" type="xsd:boolean"
        use="optional" default="false"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="StringPatternConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:sequence>
        <xsd:element name="pattern" type="xsd:string"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="MultiValueRangeConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:attribute name="min-values" type="xsd:nonNegativeInteger"
        use="optional" default="0"/>
      <xsd:attribute name="max-values" type="xsd:nonNegativeInteger"
        use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="UniqueConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:attribute name="case-sensitive" type="xsd:boolean"
        use="optional" default="true"/>
      <xsd:attribute name="type" type="xsd:string" use="optional"
        default="varying-temporal"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="ValidValueConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:sequence>
        <xsd:element name="values">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="value" minOccurs="1"
                maxOccurs="unbounded">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:choice>
                      <xsd:element name="string-value"
                        type="xsd:string"/>
                      <xsd:element name="integer-value"
                        type="xsd:long"/>
                      <xsd:element name="decimal-value"
                        type="xsd:double"/>
                      <xsd:element name="boolean-value"
                        type="xsd:boolean"/>
                    </xsd:choice>
                    <xsd:element name="label"
                      type="xsd:string"/>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="NonNullConstraint">
  <xsd:complexContent>
    <xsd:extension base="ValueConstraint">
      <xsd:attribute name="deferred" type="xsd:boolean" default="true"
        use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

</xsd:schema>
```

A.2 Model Definition

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:t="http://xmlns.oracle.com/iam/rm/type/def/1_0"
  xmlns:c="http://xmlns.oracle.com/iam/rm/cache/config/1_0"
  xmlns="http://xmlns.oracle.com/iam/rm/temporal/config/1_0"
  targetNamespace="http://xmlns.oracle.com/iam/rm/temporal/config/1_0"
  attributeFormDefault="unqualified"
  elementFormDefault="qualified" >

  <xsd:import namespace="http://xmlns.oracle.com/iam/rm/type/def/1_0"/>
  <xsd:import namespace="http://xmlns.oracle.com/iam/rm/cache/config/1_0"/>

  <xsd:element name="temporal" type="TemporalType"/>

  <xsd:simpleType name="dbEntityName">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[a-zA-Z][a-zA-Z0-9_]*"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:complexType name="TemporalType">
    <xsd:sequence>
      <xsd:element name="compatible-versions" type="CompatibleVersions"
        minOccurs="0"/>
      <xsd:element name="dependencies" type="Dependencies" minOccurs="0"/>
      <xsd:element name="default-cache-config" type="c:cacheConfigType"
        minOccurs="0"/>
      <xsd:element name="domain-definitions" minOccurs="0" maxOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="domain-definition"
              type="DomainDefinition" minOccurs="1"
              maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="permission-definitions" minOccurs="0"
        maxOccurs="1">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="permission-definition"
              type="PermissionDefinition" minOccurs="1"
              maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="system-indexes" minOccurs="0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="index" type="IndexDefinition"
              minOccurs="1" maxOccurs="unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

<xsd:element name="type-definitions" minOccurs="0" maxOccurs="1">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="structural-type-definition"
        type="StructuralTypeDefinition"/>
      <xsd:element name="mixin-type-definition"
        type="MixinTypeDefinition"/>
      <xsd:element name="relationship-type-definition"
        type="RelationshipTypeDefinition"/>
      <xsd:element name="extension-type-definition"
        type="ExtensionTypeDefinition"/>
      <xsd:element name="hierarchy-type-definition"
        type="HierarchyTypeDefinition"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
<xsd:element name="system-constraints" type="ConstraintList"
  minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="id" type="xsd:string" use="required"/>
<xsd:attribute name="version" use="required" type="t:Version"/>
<xsd:attribute name="previous-version" use="optional" type="t:Version"/>
</xsd:complexType>

<xsd:complexType name="CompatibleVersions">
  <xsd:sequence>
    <xsd:element name="compatible-version" type="CompatibleVersion"
      minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CompatibleVersion">
  <xsd:attribute name="version" type="t:Version" use="required"/>
</xsd:complexType>

<xsd:complexType name="Dependencies">
  <xsd:sequence>
    <xsd:element name="dependency" type="Dependency" minOccurs="1"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="Dependency">
  <xsd:attribute name="id" type="xsd:string" use="required"/>
  <xsd:attribute name="version" type="t:Version" use="required"/>
</xsd:complexType>

<xsd:complexType name="ConstraintList">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="mutually-exclusive-set-constraint">
      <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="1">
          <xsd:element name="attributes" minOccurs="0"
            maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:sequence minOccurs="2" maxOccurs="unbounded">
                <xsd:element name="attribute"
                  type="dbEntityName" />
              </xsd:sequence>
            </xsd:complexType>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:choice>
  </xsd:complexType>

```

```

        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="custom-constraint">
    <xsd:complexType>
        <xsd:sequence minOccurs="0" maxOccurs="1">
            <xsd:element name="body" type="xsd:string"/>
        </xsd:sequence>
        <xsd:attribute name="name" type="dbEntityName"
            use="required"/>
    </xsd:complexType>
</xsd:element>
</xsd:choice>
</xsd:complexType>

<xsd:complexType name="DomainDefinition">
    <xsd:group ref="t:PrimitiveTypeChoice"/>
    <xsd:attribute name="name" type="dbEntityName" use="required"/>
</xsd:complexType>

<xsd:complexType name="PermissionDefinition">
    <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="description" type="xsd:string"/>
        <xsd:element name="privilege-name" type="xsd:string"/>
        <xsd:element name="privilege-description" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
</xsd:complexType>

<xsd:complexType name="StructuralTypeDefinition">
    <xsd:complexContent>
        <xsd:extension base="EntityTypeDefinition">
            <xsd:sequence>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="RelationshipTypeDefinition">
    <xsd:complexContent>
        <xsd:extension base="TypeDefinition">
            <xsd:sequence>
                <xsd:element name="primary-key-definitions">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="primary-key-definition"
                                type="PrimaryKeyDefinition" minOccurs="2"
                                maxOccurs="unbounded"/>
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element name="bypassing-relationship-paths"
            minOccurs="0">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="bypassing-relationship-path"
                        minOccurs="1" maxOccurs="2">
                        <xsd:complexType>
                            <xsd:sequence>
                                <xsd:element name="title"
                                    type="xsd:string"/>
                            </xsd:sequence>
                            <xsd:attribute name="id" type="xsd:string"
                                use="required"/>
                            <xsd:attribute name="local-attribute"
                                type="dbEntityName" use="required"/>
                            <xsd:attribute name="foreign-attribute"
                                type="dbEntityName" use="required"/>
                        </xsd:complexType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="MixinTypeDefinition">
    <xsd:complexContent>
        <xsd:extension base="EntityTypeDefinition">
            <xsd:sequence>
                <xsd:element name="related-types">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="related-type"
                                type="dbEntityName" minOccurs="1"
                                maxOccurs="unbounded" />
                        </xsd:sequence>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

<xsd:complexType name="EntityTypeDefinition">
    <xsd:complexContent>
        <xsd:extension base="TypeDefinition">
            <xsd:attribute name="extends" type="dbEntityName" use="optional"/>
            <xsd:attribute name="abstract" type="xsd:boolean" use="optional"
                default="false"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:complexType name="TypeDefinition">
  <xsd:sequence>
    <xsd:element name="title" type="xsd:string" nillable="false"/>
    <xsd:element name="title-attribute" type="xsd:string" minOccurs="0"/>
    <xsd:element name="cache-config" type="c:cacheConfigType"
      minOccurs="0"/>
    <xsd:element name="snapshot-interface-classname" type="xsd:string"
      minOccurs="0"/>
    <xsd:element name="index-definitions" minOccurs="0">
      <xsd:complexType>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="index" type="IndexDefinition"/>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="attribute-definitions" minOccurs="0">
      <xsd:complexType>
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
          <xsd:element name="reference-attribute"
            type="ReferenceAttributeDefinition"/>
          <xsd:element name="attribute" type="AttributeDefinition"/>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="related-audit-objects" type="RelatedAuditObjects"
      minOccurs="0"/>
    <xsd:element name="permissions" type="Permissions" minOccurs="0"/>
    <xsd:element name="access-policy" type="AccessPolicy" minOccurs="0"/>
    <xsd:element name="entity-constraints" type="ConstraintList"
      minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="dbEntityName" use="required"/>
  <xsd:attribute name="disjointed" type="temporalConstraintType"/>
  <xsd:attribute name="overlapping" type="temporalConstraintType"/>
  <xsd:attribute name="continuous" type="temporalConstraintType"/>
  <xsd:attribute name="storage-type" type="storageType"
    default="pan-temporal" use="optional"/>
  <xsd:attribute name="definition" type="definitionType" default="internal"
    use="optional"/>
</xsd:complexType>

<xsd:simpleType name="temporalConstraintType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="true"/>
    <xsd:enumeration value="fill"/>
    <xsd:enumeration value="false"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="definitionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="external"/>
    <xsd:enumeration value="internal"/>
  </xsd:restriction>
</xsd:simpleType>

```

```

<xsd:simpleType name="storageType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="temporal"/>
    <xsd:enumeration value="current-time"/>
    <xsd:enumeration value="pan-temporal"/>
    <xsd:enumeration value="historical"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="HierarchyTypeDefinition">
  <xsd:sequence>
    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="node-definitions">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="node-definition" minOccurs="1"
            maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="node-relationships"
                  minOccurs="0">
                  <xsd:complexType>
                    <xsd:sequence>
                      <xsd:element
                        name="node-relationship"
                        minOccurs="1"
                        maxOccurs="unbounded">
                        <xsd:complexType>
                          <xsd:attribute
                            name="foreign-type"
                            type="dbEntityName"
                            use="required"/>
                          <xsd:attribute
                            name="relationship-path"
                            type="dbEntityName"
                            use="required"/>
                        </xsd:complexType>
                      </xsd:element>
                    </xsd:sequence>
                  </xsd:complexType>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="default-parent"
            minOccurs="0">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element
                  name="identifying-attributes"
                  type="AttributeValueList"/>
              </xsd:sequence>
              <xsd:attribute name="object-type"
                type="dbEntityName" use="required"/>
              <xsd:attribute
                name="referring-attribute"
                type="dbEntityName" use="required"/>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```

        <xsd:attribute name="object-type"
            type="dbEntityName" use="required"/>
        <xsd:attribute name="anchor" type="xsd:boolean"
            use="optional" default="false"/>
        <xsd:attribute name="mandatory" type="xsd:boolean"
            use="optional" default="false"/>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="default-anchors" minOccurs="0">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="default-anchor" maxOccurs="unbounded">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name="identifying-attributes"
                            type="AttributeValueList"/>
                        <xsd:element name="attributes"
                            type="AttributeValueList" minOccurs="0"/>
                    </xsd:sequence>
                    <xsd:attribute name="object-type"
                        type="dbEntityName" use="required"/>
                    <xsd:attribute name="hierarchy-root-attribute"
                        type="dbEntityName" use="optional"/>
                </xsd:complexType>
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="permissions" type="Permissions" minOccurs="0"/>
</xsd:sequence>
<xsd:attribute name="id" type="dbEntityName" use="required"/>
<xsd:attribute name="single-anchor" type="xsd:boolean" use="optional"
    default="false"/>
</xsd:complexType>

<xsd:complexType name="AttributeValueList">
    <xsd:sequence>
        <xsd:element name="attribute" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:choice>
                    <xsd:element name="integer-value" type="xsd:long"/>
                    <xsd:element name="decimal-value" type="xsd:double"/>
                    <xsd:element name="datetime-value" type="t:dateTime"/>
                    <xsd:element name="string-value" type="xsd:string"/>
                    <xsd:element name="boolean-value" type="xsd:boolean"/>
                </xsd:choice>
                <xsd:attribute name="name" type="xsd:string" use="required"/>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>

```

```

<xsd:complexType name="ExtensionTypeDefinition">
  <xsd:sequence>
    <xsd:element name="title-attribute" type="xsd:string" minOccurs="0"/>
    <xsd:element name="cache-config" type="c:cacheConfigType"
      minOccurs="0"/>
    <xsd:element name="attribute-definitions" minOccurs="0">
      <xsd:complexType>
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
          <xsd:element name="reference-attribute"
            type="ReferenceAttributeDefinition"/>
          <xsd:element name="attribute" type="AttributeDefinition"/>
          <xsd:element name="attribute-ext"
            type="AttributeExtension"/>
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="related-audit-objects" type="RelatedAuditObjects"
      minOccurs="0"/>
    <xsd:element name="permissions" type="Permissions" minOccurs="0"/>
    <xsd:element name="access-policy" type="AccessPolicy" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="extending-type-id" type="dbEntityName"
    use="required"/>
</xsd:complexType>

<xsd:complexType name="Permissions">
  <xsd:sequence>
    <xsd:element name="permission" minOccurs="1" maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:attribute name="id" type="xsd:string" use="required"/>
        <xsd:attribute name="soc-policy" use="optional"
          default="children">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="none"/>
              <xsd:enumeration value="children"/>
              <xsd:enumeration value="ancestors"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:attribute>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="RelatedAuditObjects">
  <xsd:sequence>
    <xsd:element name="related-audit-object" minOccurs="1"
      maxOccurs="unbounded">
      <xsd:complexType>
        <xsd:attribute name="incoming-relationship-path"
          type="dbEntityName" use="required"/>
        <xsd:attribute name="foreign-object-type" type="dbEntityName"
          use="required"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```
<xsd:complexType name="PrimaryKeyDefinition">
  <xsd:sequence>
    <xsd:element name="title" type="xsd:string" nillable="false"/>
    <xsd:element name="outgoing-relationship-path-title" type="xsd:string"
      nillable="false"/>
    <xsd:element name="incoming-relationship-path-title" type="xsd:string"
      nillable="false"/>
    <xsd:element name="getter" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="dbEntityName" use="required"/>
  <xsd:attribute name="foreign-type" type="dbEntityName" use="required"/>
  <xsd:attribute name="outgoing-relationship-path" type="xsd:string"
    use="required"/>
  <xsd:attribute name="incoming-relationship-path" type="xsd:string"
    use="required"/>
  <xsd:attribute name="parent-action" use="optional" default="restrict">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="restrict"/>
        <xsd:enumeration value="cascade"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
</xsd:complexType>

<xsd:complexType name="ReferenceAttributeDefinition">
  <xsd:sequence>
    <xsd:element name="title" type="xsd:string" nillable="false"/>
    <xsd:element name="outgoing-relationship-path-title" type="xsd:string"
      nillable="false"/>
    <xsd:element name="incoming-relationship-path-title" type="xsd:string"
      nillable="false"/>
    <xsd:element name="reference-ext" type="t:ReferenceTypeExt"
      minOccurs="0"/>
    <xsd:element name="snapshot-getter" type="xsd:string" minOccurs="0"/>
    <xsd:element name="snapshot-setter" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
  <xsd:attribute name="foreign-type" type="xsd:string" use="required"/>
  <xsd:attribute name="parent-action" use="optional" default="set null">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="restrict"/>
        <xsd:enumeration value="cascade"/>
        <xsd:enumeration value="set null"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="outgoing-relationship-path" type="xsd:string"
    use="required"/>
  <xsd:attribute name="incoming-relationship-path" type="xsd:string"
    use="required"/>
  <xsd:attribute name="multi-value" type="xsd:boolean" use="optional"
    default="false"/>
</xsd:complexType>
```

```
<xsd:complexType name="AttributeDefinition">
  <xsd:sequence>
    <xsd:element name="title" type="xsd:string" nillable="false"/>
    <xsd:group ref="t:PrimitiveTypeExtChoice" minOccurs="0"/>
    <xsd:element name="snapshot-getter" type="xsd:string" minOccurs="0"/>
    <xsd:element name="snapshot-setter" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="domain" type="xsd:string" use="required"/>
  <xsd:attribute name="searchable" type="xsd:boolean" use="optional" />
</xsd:complexType>

<xsd:complexType name="IndexDefinition">
  <xsd:attribute name="body" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="AttributeExtension">
  <xsd:sequence>
    <xsd:group ref="t:PrimitiveTypeExtChoice" minOccurs="1"/>
  </xsd:sequence>
  <xsd:attribute name="extending-domain" use="required"/>
</xsd:complexType>

<xsd:complexType name="AccessPolicy">
  <xsd:attribute name="read-permission" type="xsd:string" use="optional"/>
  <xsd:attribute name="read-audit-details-permission" type="xsd:string"
    use="optional"/>
</xsd:complexType>
</xsd:schema>
```

Index

A

abstractIdentity object type, 1-15
abstractOrg object type, 1-22
abstractRole object type, 1-4
access control, 1-34
accessibility, 0-xi
approverRole object type, 1-6
attribute definitions, about, 1-2
attributes
 adding, 2-2
 modifying, 2-3
audit system privilege, about, 1-3
auditEvent object type, 1-42
auditEventDetail object type, 1-43

B

baseBundle object type, 1-44
building object type, 1-24
bundling configurations for deployment, 2-8, 2-9
businessRole object type, 1-6, 1-7

C

class loading order, configuring in WebSphere, 3-6
colons, as delimiters in CAR collections, 2-9
configuration file archives (CAR), 2-9
configuration object type, 1-44
configurations.car file, extracting files from, 2-2, 3-2
costCenterHierarchy object type, 1-33
country object type, 1-24
customization
 approaches to, 2-1
 data model, 2-7
 version dependencies, 2-7
customizations
 deploying data model, 2-9
 deploying UI, 3-4
 reverting, 2-1

D

data model
 concepts, 1-1
 deploying customizations, 2-9
 layering, about, 1-1

database properties file, for manual deployment and
 other commands, 2-9
dcObject object type, 1-25
delegate system permission, about, 1-3
delimiters in CAR collections, 2-9
dependencies, version, 2-7
division object type, 1-26
domain definitions, about, 1-2
domains, defined, 2-2

E

entitlements
 about, 1-40
 itPrivilege entitlement object, 1-40

F

floor object type, 1-26
foreign-type reference component, defined, 2-5

G

globalheader element, 3-3
grant system permission, about, 1-3

H

hierarchies
 about, 1-31
 costCenterHierarchy object type, 1-33
 locationHierarchy object type, 1-34
 reporting hierarchy, 1-33
hierarchy object type, 1-32

I

id reference component, defined, 2-4
image files, 3-3
incoming-relationship-path reference component,
 defined, 2-5
internal security, 1-34
itPrivilege object type, 1-40
itRole object type, 1-9
itRolePrivilegeMapping object type, 1-41

L

locality object type, 1-27
localizedBundle object type, 1-44
locationHierarchy object type, 1-34
logos, 3-3

M

manage system permission, about, 1-3

O

oimSystem system user, 1-20
organization object type, 1-28
ou object type, 1-29
outgoing-relationship-path reference component,
defined, 2-5

P

parent-action reference component, defined, 2-5
permissions
system, 1-2, 1-35
person object type, 1-16
pluginPack object type, 1-45
primordial data model, about, 1-1
privileges, 1-36

R

reference attributes
adding, 2-5
defined, 2-4
values for components in, 2-4
relationship attributes, adding, 2-5
relevantRoleAttribute object type, 1-13
reportingHierarchy object type, 1-33
resource properties, 3-2
reverting customizations, 2-1
roleGrant object type, 1-13
roleMapping object type, 1-13
room object type, 1-30
row count, changing in search component, 3-4

S

sample XML files for configuration, 2-2
search component, changing row count, 3-4
semicolons, as delimiter in CAR collections, 2-9
sphere of control, about, 1-31
standard configuration, 2-2
standard data model, about, 1-1
standard.xml file, 2-2
structural type objects
adding, 2-6
defined, 2-6
style.css file, 3-2
sysPermissionAssociation object type, 1-39
sysRolePrivilegeMapping object type, 1-39
system access control, 1-34

system administrator, 1-20
system privileges, about, 1-2
system user, 1-19
systemIdentity object type, 1-19
systemPermission object type, 1-35
systemPrivilege object type, 1-36
systemResource object type, 1-37
systemResourceType object type, 1-38
systemRole object type, 1-11

T

TTY access, 0-xi

U

uerRoleAssignment object type, 1-14
UI customizations, deploying, 3-4
unassigned objects, 1-33

V

version dependencies, 2-7

W

webui.ear file, deploying on WebSphere, 3-5
webui.war file
deploying on JBoss, 3-5
deploying on WebLogic, 3-5

X

XML files, for configuration, 2-2