Oracle Access Governance

Oracle Access Governance

Release Latest

April 2025

ORACLE

Oracle Access Governance Oracle Access Governance, Release Latest

Contents

1 Getting Started

1-1
About Access Governance 1-1
1-1
Access Reviews 1-1
Access Review Campaigns 1-2
Access Controls 1-2
Approval Workflows 1-2
Centralized Enterpise-wide Visibility into Access Profiles 1-3
Correlation 1-3
Data Transformation 1-3
Event Data Publishing 1-4
Identity Intelligence 1-4
Identity Orchestration 1-4
Identity Orchestration with JML Support 1-5
Micro-Certifications: Event-Based and Time-Driven Access Reviews 1-5
Reporting and Analytics 1-5
Self Service 1-6
User Interface 1-6
Oracle Access Governance Licensing Information 1-6
Licensing Models, Unit Metrics and SKUs 1-6
Licensing Models and Offerings 1-7
Choosing the Right Licensing Model 1-8
1-9
Oracle Cloud Infrastructure Services 1-9
Typical Workflow for Administrators 1-9
1-11
About Setting Up Users and Groups 1-11
Use Identity Domains to Onboard Users and Groups for Oracle Access Governance 1-12
Use Oracle Identity Cloud Service to Onboard Users and Groups for Oracle Access Governance 1-13
Assign Access Governance Application Roles to Users and Groups 1-14
Application Roles 1-16
About Application Roles in Oracle Access Governance 1-16

Administrator (AG_Administrator) 1-16
Service Desk Administrator (AG_ServiceDesk_Admin) 1-17
Campaign Administrator (AG_CampaignAdmin) 1-17
Enterprise-wide Browser Access Administrator (AG_Enterprise_Wide_Access_Admin) 1-18
Application Owner Administrator (AG_AppOwner_Admin) 1-18
Application Owner Restricted Administrator (AG_AppOwner_Admin_Restricted) 1-19
Access Control Administrator (AG_AccessControl_Admin) 1-20
Access Control Restricted Administrator (AG_AccessControl_Admin_Restricted) 1-20
Auditor (AG_AUDITOR) 1-21
User (AG_USER) 1-21
Application Roles and Responsibilities Reference 1-21
Predefined Application Roles 1-22

2 What's New

What's new in Oracle Access Governance 2-1
Recent Updates in Oracle Access Governance 2-1

3 Administer

System Administration 3-1
3-1
Prerequisites 3-1
Create Service Instance 3-3
Verify Service Instance 3-4
3-5
Review Service Instance 3-5
Launch Service Home Page 3-6
Edit Service Instance 3-6
Delete Service Instance 3-7
3-8
Configure Notification Email Display Name Using OCI Console 3-8
Configure an OCI Email Delivery Service for Notifications 3-9
Configure Global Settings for Notifications 3-9
Configure Notification Types 3-10
Configure Identities or Email for Sending Orchestrated System Related Notifications 3-11
Service Administration 3-11
3-11
Navigate to Manage Identities 3-13
Select Identities for Activation 3-13
Select Consumer Users 3-14

Create and Manage Organizations 3-15
Manage Account Lifecycle with Oracle Access Governance Service Desk Administrator Support 3-18
Manage Identity Attributes 3-22
Overview 3-22
View Attributes 3-23
Manage Core Attribute Settings 3-24
Manage Custom Attribute Settings 3-25
Fetch Latest Custom Attributes 3-26
3-26
Match Unmatched Accounts to an Identity 3-27
Remove Unmatched Account 3-28
3-28
Normalise Key-Values Across Orchestrated Systems 3-28
Create a Global Key-Value 3-29
View Details of a Global Key-Value 3-29
Edit Details of a Global Key-Value 3-30
Delete a Global Key-Value 3-30

4 Access Controls

Account Lifecycle Management - Automated Provisioning for Joiners Movers and Leavers (JML) Process 4-1
Employees Onboarding - Joiners Provisioning 4-2
Employee Transfers - Movers Provisioning 4-3
Employees Offboarding - Leavers De-Provisioning 4-4
4-5
Navigate to Identity Collections 4-5
Add Details 4-6
Add Primary and Additional Owners 4-6
Add system 4-6
Select Identities 4-7
Review and Submit 4-8
4-9
View and Manage Identity Collections 4-9
4-11
Create a Policy 4-11
Edit a Policy 4-12
Delete a Policy 4-12
4-13
Create a Role 4-13
Edit a Role 4-15
Delete a Role 4-15

Create an Access Bundle 4-16
Overview 4-16
Navigate to Access Bundle 4-16
Bundle Settings 4-16
Select Permissions 4-17
Add Primary and Additional Owners 4-17
Add Details 4-18
Review and Submit 4-18
4-19
View and Manage Access Bundles 4-19
4-20
Navigate to Approval Workflow 4-21
Create Approval Workflow 4-21
Build Approvals 4-21
Add Details 4-24
Add Primary and Additional Owners 4-25
Review and Submit 4-25
4-25
View Workflow Details 4-26
Edit Workflow Details 4-26
Delete Workflow 4-26
Disable Workflow 4-26

5 Integrate

5-1
Integrations in Oracle Access Governance 5-1
Identity Orchestration Functional Overview 5-2
5-3
Access Governance Agent 5-4
Data Transformations 5-4
Matching Rules 5-8
Orchestrated System Resources 5-8
5-9
Identity Orchestration Process Flow 5-9
5-10
Register and Download the Oracle Access Governance Agent 5-10
Prerequisites 5-11
Sizing Virtual Machine/Host 5-12
Install Oracle Access Governance Agent 5-12
Verify Agent 5-13
Agent Example Usage 5-14

Custom Jar Support 5-16
Agent Management Operations 5-17
Agent Parameters 5-18
Tuning Runtime Configuration 5-19
Troubleshooting Oracle Access Governance Agent 5-20
5-21
Add an Orchestrated System 5-21
Manage an Orchestrated System 5-21
Manage Orchestrated System Resources 5-22
Initiate Data Load 5-23
View Activity Log 5-23
Disable an Orchestrated System 5-24
5-24
Modify Integration Settings for an Orchestrated System 5-24
Modify Account Settings for an Orchestrated System 5-25
Configure Data Load Schedule Settings for Orchestrated Systems 5-26
Add Primary and Additional Owners 5-27
Configure Identities or Email for Sending Orchestrated System Related Notifications 5-27
Match Identity and Account Attributes using Correlation Rules 5-28
Apply Inbound Transformations for Identity and Account Attributes 5-29
Apply Outbound Transformations for Identity Attributes 5-29
5-30
Supported Systems 5-30
5-33
Authoritative Source Identity Object Attributes for Outbound Transformation 5-34
Authoritative Source Identity Object Attributes for Inbound Transformation and Identity Attributes Customization 5-36
Managed Systems Account Object Attributes for Inbound Transformation 5-40
Custom User and Account Attributes 5-43
Examples for Outbound Data Transformation 5-44
Examples for Inbound Data Transformation and Identity Attributes 5-47
Oracle Access Governance Integration Functional Overview: Supported Operations in Orchestrated System 5-50
Configure Orchestrated System 5-50
Load Data 5-50
Create Account 5-50
Assign Permissions 5-51
Remove Permissions 5-51
Change Password 5-51
Revoke Account 5-52
Enable Account 5-52
Integrate with Orchestrated Systems 5-52

5-52
Preinstall 5-52
Set Up Oracle Identity Governance Integration 5-53
Supported Attributes for User Data Load Filtering 5-57
Database Setup Steps for Event-driven Data Load 5-59
Configure Integration Between Oracle Access Governance and Oracle Cloud Infrastructure (OCI) 5-61
Set Up Identity Resources on OCI to Connect to Oracle Access Governance 5-61
Establish Connection by Adding a New Orchestrated System - OCI IAM 5-66
Supported Operations for OCI 5-68
5-69
Prerequisites 5-70
Configure 5-72
Post Configuration 5-75
Database Application Tables 5-76
5-76
5-79
5-89
5-99
5-108
5-118
5-129
5-149
5-167
5-187
Preinstall 5-187
Install 5-188
Postinstall 5-191
5-191
Preinstall 5-191
Install 5-192
Postinstall 5-194
5-195
Preinstall 5-195
Install 5-197
5-198
Preinstall 5-198
Install 5-200
Postinstall 5-203
5-203
Prerequisites 5-203
Configure 5-208

Postinstall 5-212
5-212
Prerequisites 5-212
Configure 5-217
Post Configuration 5-219
5-219
Prerequisites 5-220
Configure 5-220
Post Configuration 5-223
5-223
Prerequisites 5-223
Configure 5-224
Post Configuration 5-227
Oracle Fusion Cloud Applications 5-227
5-227
5-230
5-234
Oracle Health EHR (formerly Cerner Millenium) 5-237
Overview: Integrate Oracle Access Governance with Oracle Health EHR (formerly Cerner Millenium) 5-237
Oracle Health EHR (formerly Cerner Millenium) Integration Architecture Overview 5-238
Oracle Health EHR (formerly Cerner Millenium) Integration Functional Overview 5-238
Oracle Health EHR (formerly Cerner Millenium) Components Certified for Integration with Oracle Access Governance 5-240
Supported Configuration Modes for Oracle Health EHR (formerly Cerner Millenium) 5-240
Supported Operations When Provisioning To Oracle Health EHR (formerly Cerner Millenium) 5-240
Default Supported Attributes 5-241
Troubleshooting 5-242
PeopleSoft 5-242
5-242
5-246
5-250
SAP Ariba 5-257
Integrate Oracle Access Governance with 5-257
Configure Integration Between Oracle Access Governance and 5-260
Integration Reference 5-266
SAP S4HANA 5-267
Integrate Oracle Access Governance with 5-267
Configure Integration Between Oracle Access Governance and 5-269
Integration Reference 5-274
SAP SuccessFactors 5-275
5-275

5-276
5-280
5-361
Prerequisites 5-361
Configure 5-362
Post Configuration 5-365
5-365
Prerequisites 5-365
Configure 5-366
Post Configuration 5-369
5-369
Prerequisites 5-369
Configure 5-370
Post Configuration 5-372
5-373
Prerequisites 5-373
Configure 5-374
Post Configuration 5-377
Integrate with Microsoft Entra ID 5-377
Prerequisites 5-377
Configure 5-380
Post Configuration 5-384
5-384
Prerequisites 5-384
Configure 5-385
Postconfiguration 5-387
5-387
Prerequisites 5-387
Configure 5-389
Post Configuration 5-392
Generic REST 5-393
5-393
5-399
5-416
Arcon Privileged Access Management (Arcon PAM) 5-447
5-447
5-448
5-451

6 Who Has Access to What

Who Has Access to What: Comprehensive Access Profile Visibility across Enterprises 6-1

Why Enterprise-wide Access Visibility Matters? 6-1
Browsing Views: Selection Parameters to Explore Access Profile across Enterprise 6-2
Search Capabilities: Using Keywords, Suggested and Advanced Filters 6-4
Usage Examples: Monitoring Access Profile Details in an Enterprise 6-4
Access Reviews in Enterprise-wide Browser 6-5
Explore Access Profile in an Enterprise 6-7
View Access Profile across Enterprise 6-7
Search and Apply Filters 6-8
Manage Access Profile Layout 6-9
Generate User-Created Access Reviews 6-10
My Directs' Access - View Access Profile Information for Team 6-11
Enterprise-wide Access Profile Reference 6-11
Identities 6-11
Identity Collections 6-12
Organizations 6-14
Permissions 6-14
Policies 6-18
Resources 6-19
Roles 6-20

7 Access Reviews

Access Reviews in Oracle Access Governance - Certify Access Privileges with Campaigns and Event-Driven Micro Certifications 7-1
Key Benefits of performing Access Reviews with Oracle Access Governance 7-1
Types of Access Reviews Offered by Oracle Access Governance 7-2
Identity Access Reviews 7-2
Identity Access Reviews based on Permissions Assigned Directly in Managed Systems 7-3
Policy Reviews 7-5
Identity Collection Reviews 7-5
Resource Ownership Reviews 7-5
Event-Based Micro-Certifications 7-6
Eligible Orchestrated System Types to Launch Access Review Campaigns 7-6
Review Access to Systems Managed by Oracle Access Governance 7-6
Review Accesses for Cloud Services Managed by Oracle Cloud Infrastructure (OCI) 7-8
Review Access to Systems Managed by Oracle Identity Governance (OIG) 7-9
Usage Examples: Certifying Access Privileges with Access Review Campaigns and Event-based Reviews 7-9
Access Review Campaigns 7-10
Working with Access Review Campaigns 7-10
Access Review Campaign Stages 7-11
Understanding Self-Certification Guardrails 7-13

Understanding Fallback Mechanism: Methods to Prevent Campaign Termination 7-13
Best Practices: Guidelines to Consider While Working With Campaigns 7-15
Create Identity Access Review Campaigns 7-16
Prerequisites 7-16
Navigate to Campaigns 7-16
Select Criteria for your Access Reviews 7-17
Add Access Reviewers by Selecting Approval Workflow 7-18
Add Campaign Details 7-19
Review and Submit the Campaign 7-19
Create Policy Review Campaigns 7-19
Navigate to Campaigns 7-20
Select Criteria for your Access Reviews 7-20
Add Access Reviewers by Selecting Approval Workflow 7-20
Add Campaign Details 7-21
Review and Submit the Campaign 7-21
Certify Group Memberships with Identity Collections Review Campaigns 7-22
Navigate to Campaigns 7-22
Select Criteria for your Access Reviews 7-22
Add Access Reviewers by Selecting Approval Workflow 7-23
Add Campaign Details 7-23
Review and Submit the Campaign 7-24
Create Ownership Review Campaigns 7-24
Navigate to Review Ownership Campaigns 7-24
Choose Resources for Ownership Reviews 7-24
Apply Filters to Select Resources 7-25
Add Access Reviewers by Selecting Approval Workflow 7-25
Add Campaign Details 7-26
Review and Submit the Campaign 7-26
Manage and Monitor Access Review Campaigns 7-26
Search and Apply Filters to View Available Campaigns 7-27
View Campaign Details 7-27
Edit an Access Review Campaign 7-28
Clone a Campaign 7-28
Approve a Campaign 7-29
Terminate a Campaign 7-29
View and Download Access Review Reports 7-30
Event-Based Micro Certifications 7-30
Micro-Certifications: Event Driven Access Reviews 7-30
Change Event 7-31
Timeline Event 7-32
Unmatched Accounts Event 7-32
Configure and Manage Event-based Access Reviews 7-32

Configure Change Event Access Review 7-32
Configure Shared Workflow for Multiple Change Events 7-33
Configure Timeline Event Access Reviews 7-34
Configure Unmatched Accounts Access Review 7-35
View Event Details 7-36
Edit Event-based Access Reviews 7-36
Delete Event Type for Access Reviews 7-36
7-37
Navigate to the Event-Based Access Reviews Report Service 7-37
Run Event-Based Access Reviews Report 7-37
View Event-Based Access Reviews Report Results 7-38
Additional Actions 7-39
Perform Access Reviews 7-39
Understanding Reviewer's Actions for Effective Access Certification 7-39
Access Review Task Types in Oracle Access Governance 7-40
Intelligent Insights - Review Recommendations based on Prescriptive Analytics 7-41
Audit Trail: Monitoring Access Review and Access Request Decisions 7-41
Recent Change Events Log: Tracking Attribute Changes 7-42
Delegating your Review Tasks 7-42
Reassigning a Review Task 7-42
Bulk Changes - Managing Multiple Review Items Simultaneously 7-43
Perform Access Reviews - Evaluate and Certify Access Review Tasks 7-43
Review Identity Access Tasks 7-43
Review Policy and Identity Collection with Access Control Tasks 7-44
Review Unmatched Accounts with Ownership Tasks 7-46
Review Resource Ownership with Ownership Task 7-47
Reassign a Review Task 7-47

8 Self Service

View Access Details and Manage Account 8-1
Identities 8-1
Change Account Password 8-2
8-2
Manage Approvals 8-3
Preventive Segregation of Duties (SOD) Analysis 8-3
8-4
View My Access Requests 8-5
8-6
Request Access to a Resource 8-6
Request Access To A Role 8-7
8-7

Set up Your Delegation Preferences 8-8
Edit a Delegation 8-9
Delete a Delegation 8-10

9 Data Feed

Event Data Publisher in Oracle Access Governance 9-1
Understanding Data Event Publishing Flow 9-1
Initial Data Event and Incremental Data Events : Day 0 and Day N Events 9-2
Available Data Components for Publishing 9-3
Configure Event Data Publisher in Oracle Access Governance 9-4
Prerequisites 9-4
Set Up OCI Tenancy for Data Event Publisher 9-5
Configure Settings for Data Publisher in Oracle Access Governance 9-9
Event Data Publishing Reference Schema and Sample Files 9-10
Header Schema and Sample Output Reference 9-10
Header Schema Attribute Definition 9-14
Identity Reference Schema and Sample Output File 9-15
Identity Schema Attribute Definition 9-21
Identity Collection Reference Schema and Sample Output File 9-21
Identity Collection Schema Attribute Definition 9-26
Policies Reference Schema and Sample Output File 9-27
Policies Schema Attribute Definition 9-29
Resource Reference Schema and Sample 9-30
Resources Schema Attribute Definition 9-32
Resource to Policy Statement 9-32
Resources to Policy Schema Attribute Definition 9-33
Policy Statement to Resource 9-34
Policy to Resources Schema Attribute Definition 9-36

10 Reference

10-1
A 10-1
C 10-1
D 10-2
E 10-2
G 10-2
I 10-2
J 10-3
M 10-3
N 10-3

O 10-4
P 10-4
R 10-4
S 10-4
W 10-5
Supported Languages in Oracle Access Governance 10-5
Language and Locale 10-5
Switching to the Preferred Language in your Browser 10-6

1

Getting Started

About Access Governance

Oracle Access Governance is a cloud-native Identity Governance and Administration (IGA) solution that provides insights-based access reviews, identity analytics, and intelligence capabilities for businesses.

Oracle Access Governance provides features including:

- Visibility of enterprise compliance by providing details on who has access to what.
- Ability for reviewers to optimize user privileges through intelligent access review campaigns.
- Actionable identity intelligence by building deep insights into potential security violations that enable rapid remediation of identity and access challenges.
- Continuous compliance to meet broader organizational needs.

Key features of Oracle Access Governance include the following:

Access Reviews

Process to evaluate and certify the access privileges granted to identities within an enterprise. It checks and certifies if privileges granted are still required and align with the current job at work. With Access Reviews, you can make swift and accurate review decisions by examining insights and AI-powered recommendations based on prescriptive analytics.

Key Functions

- Multiple types of campaigns to support periodic or ad-hoc reviews
- Configurable self-certification capability to allow self-reviews.
- Intelligent fall-back mechanism to avoid sudden termination of campaigns. It auto-assign the next applicable reviewer or campaign owner.
- Automated micro-certifications, triggered only when there are changes in the system of record, occurrence of an important date or time milestone, or detection of an orphan account.
- Review Identity, Access Control, and Ownership review tasks.
- Delegate or Reassign review tasks to other reviewer.

Access Review Campaigns

Periodic or ad hoc snapshot-based reviews, capturing all the relevant access information at a given point of time, and then assessing and generating access review tasks. It improves certification efficiency by providing actionable insights based on prescriptive analytics.

Key Functions

- Certify identity accesses and assigned privileges across all orchestrated systems connected with Oracle Access Governance.
- Certify membership in a group to verify if only eligible set of members are assigned to a group. This is commonly known as "Group membership reviews."
- Verify the principle of least-privilege by reviewing policy and policy constructs with policy reviews.
- Review accountability of resources by running resource ownership reviews.

Access Controls

Permission management and administration feature that governs how resource access is granted to identities across your enterprise or organization. With Oracle Access Governance, you can leverage the Attribute-Based Access Control (ABAC), grant access to role Role-Based Access Control (RBAC), or Policy-Based Access Control (PBAC) permission models.

Key Functions

- **Role-based access control (RBAC)**: Assign permissions to users associated with their job profile or functions.
- **Attribute-based access control (ABAC)**: Assign membership to identity collections based on core or custom identity attributes
- **Policy-based access control (PBAC)**: Assign permissions to users by defining a policy.
- Request access to roles or access bundles directly via self-service module, and processed only after approval.

Approval Workflows

Code-less workflow templates to obtain approvals for tasks in Oracle Access Governance. You can choose out-of-the-box or build your own sequential or parallel workflow paths.

- Involves code-less workflow creation.
- Intelligent fall-back mechanism to avoid sudden termination of tasks.
- Sends notification emails about assigned and pending access reviews.
- Supports complex multistage approvals through sequential or parallel workflows to meet your business needs.
- Reviewer can make decisions to accept, revoke, or reassign items in access reviews or request approval task.
- Get suggestions for the intelligent workflow based on selected criteria.

Centralized Enterpise-wide Visibility into Access Profiles

With Enterprise-wide Browser, you gain insights into access usage to detect and prevent any potential misuse. With Enterprise-wide Browser, you get comprehensive visibility on all the components, access information, and resources within an enterprise framework.

Key Functions

- Centralized dashboard with multiple browsing views to view access information. The access is presented across multiple anchoring points:
  - **Identities**: Understand who has access to what.
  - **Identity collections**: Group identities logically.
  - **Organizations**: View access within specific organizational units.
  - **Roles**: Explore role-based access.
  - **Policies**: Understand access policies.
  - **Permissions**: View details of specific permissions.
  - **Applications and resources**: Identify which applications and resources are accessed.
- Advanced search capabilities, including keyword search, suggested filters, and advanced filters to get specific and relevant results.
- Generate monthly access review report based on the date range and access review Enterprise-wide Browser. You can see breakdown of pending, approved, or revoked access review decisions for user role, user account and permission.
- Generate spontaneous user-created access reviews for a resource within Enterprise-wide Browser.

Correlation

Correlation or Matching Rules allows you to configure a set of rules to match and associate ingested identity or account to an existing identity. With this, you can leverage identity matching and account matching to build a composite identity profile. These are beneficial to automatically associate multiple accounts incoming from Managed systems with identities and avoid accumulation of unmatched accounts.

Key Functions

- Configure rules for Identity-Identity Matching for Authoritative Sources
- Configure rules for Identity-Account Matching for Managed Systems and Authoritative Sources
- Match an unmatched account with an existing identity, manually.

Data Transformation

Data Transformation feature in Oracle Access Governance allows you to modify and transform incoming identity and account data from Authoritative Source or Managed Systems, or transform outgoing data being provisioned to Managed Systems.

Key Functions

- Inbound Transformation for Identity Attributes
- Inbound Transformation on Account Attributes

- Outbound Transformation on Account Attributes using Identity Attributes
- Composite Identity Profile Transformation within Oracle Access Governance

Event Data Publishing

Export and continually publish data events in real-time to external systems using the **Data Feed** service. **Data Feed** publishes real-time updates as a continuous stream in a sequential order.

Key Functions

- Publishes real-time updates as a continuous stream in a sequential order to OCI Streams.
- Publishes the following data components to external systems:
  - All the active identities, workforce, or consumers are published as IDENTITY events to OCI Buckets and OCI streams.
  - All the available OCI IAM group ingested into Oracle Access Governance will be published as GROUP events.
  - All the available OCI policies ingested into Oracle Access Governance will be published as TARGET_ACCESS_POLICY_STATEMENT events.
  - All available resources across all orchestrated systems ingested into Oracle Access Governance will be published as RESOURCE events.
  - Access mapping for OCI policies and OCI resources.

Identity Intelligence

Oracle Access Governance analyzes each identity and its privileges, builds insights into potential high-risk assignment and security violations, and recommends remediations. This enables access reviewers to make corrective decisions quickly. This feature enables:

- Assimilation and analysis of identity data and access privileges.
- Recognition of contextual insights and identification of security blind spots.
- Remediation recommendations enable access reviewers to make corrective decisions quickly.

Identity Orchestration

Oracle Access Governance brings together diverse Authoritative Sources and Managed Systems by supporting low-code integrations. It facilitates data transformations and correlation rules which ensures data coherence. It extracts the required identity data from various systems into Oracle Access Governance and enables businesses to perform robust access control, intelligent access reviews, and perform fulfillment through account provisioning.

Key Functions

- Specialized and generic low-code integration with various on-premises systems and cloud applications and systems.
- Extract only the required information, such as identity attributes, permission assignments, and policies, into Oracle Access Governance.
- Support transformation and correlation rules for identity and account attributes, to build composite identity profile and account information.

- Process the identity data and using it for access controls, access reviews, workflows, and so on.
- Provision and synchronize data between the orchestrated systems to support Identity Lifecycle.

Identity Orchestration with JML Support

Oracle Access Governance supports creation, modification, and deletion of identity accounts and their access permissions based on attribute change in the integrated Orchestrated system. You can configure access controls to automatically provision and de-provision accounts as part of the Identity Lifecycle Management. It supports all the three stages – Joiners, Movers, and Leavers (JML).

Key Functions

- **Account Creation**: Provisioning of new accounts whenever a new user is detected. The provisioning is completed based on access request, roles, or defined policies.
- **Account Modification**: Automated modification of account attributes when identity attributes are updated in the authoritative source and synchronized. Oracle Access Governance assigns new permissions and revoke or disable unessential permissions associated with the account.
- **Account Deletion:** Permanently delete accounts when no longer required, such as when employee leaves the organization.

Micro-Certifications: Event-Based and Time-Driven Access Reviews

Event-Based Reviews are the action-oriented reviews carried out by Oracle Access Governance whenever an event, such as change event, timeline event, or unmatched account event, is detected. These generate near real-time access reviews so that prompt actions can be taken whenever these pre-defined events are detected.

Key Functions

- Launch change event type review based on update in core and custom identity attributes.
- Launch time-driven access reviews annually on a given date.
- Configure set up to trigger unmatched account event to detect orphan account.
- Define auto-actions for low-risk access reviews or unmatched accounts.
- Generate insights on event-based access reviews.

Reporting and Analytics

Oracle Access Governance enables reporting and analytics by providing various out-of-the-box summary reports and insights.

Key Functions

- 360-degree visibility into identities, accounts, policies, roles, resources and permissions in an intuitive dashboard.
- Discover, determine risk, and monitor accounts with privileged access for anomalous behavior.
- Monthly report on access reviews based on the date range and access review type or event type

Self Service

Oracle Access Governance empower users to independently request or complete routine tasks without the administrative intervention.

Key Functions

- Request access to roles or access bundles created within Oracle Access Governance
- Change account passwords for Managed Systems
- Review or approve user access permissions
- Delegate access request or access review tasks to other users.
- View your own accesses, containing details on granted roles, permissions, accounts, ownership, resources, and so on.

User Interface

The Oracle Access Governance Console provides user-friendly intuitive user interface (UI) for performing access reviews, managing access controls, carry out self-service tasks, and so on. Intelligent dashboards are available that assist in focusing on prioritized and urgent review tasks.

Oracle Access Governance Licensing Information

Oracle Access Governance offers three types of licensing models covering various service offerings based on identity type and identity segments. Let's explore various licensing types and other requirements to use Oracle Access Governance.

This Licensing Information document is a part of the product or program documentation under the terms of your Oracle services or license agreement and is intended to help you understand the program editions, entitlements, restrictions, prerequisites, special license rights, and/or separately licensed third party technology terms associated with the Oracle services or software program(s) covered by this document (the "Program(s)"). If you have a question about your rights and obligations, please contact your Oracle sales representative and/or contact the applicable Oracle License Management Services representative listed on http://www.oracle.com/us/corporate/license-management-services/index.html.

Licensing Models, Unit Metrics and SKUs

Licensing and pricing for Oracle Access Governance broadly depends on your licensing model, identity segments (tiers), and identity types.

Licensing Models in Oracle Access Governance

Oracle Access Governance offers three licensing models:

- **Oracle Access Governance for Oracle Cloud Infrastructure**: This contains integration support for Oracle Cloud Infrastructure (OCI) only.
- **Oracle Access Governance for Oracle Workloads**: This includes integration support for Oracle applications and services, including Oracle Cloud Infrastructure.
- **Oracle Access Governance Premium**: This includes integration support for all the available applications and services, including Oracle applications and services.

Workloads refer to collection of applications or services (Authoritative Source and Managed System) that can be integrated, governed and managed using Oracle Access Governance. **Oracle Workloads** refers to Oracle applications and services, such as Oracle Unified Directory, Oracle Siebel, Oracle Primavera, Oracle Identity Governance, Oracle Database User Management, and so on.

Unit Metrics and Stock Keeping Units (SKUs)

You can manage access privileges and perform various governance operations by marking identities as **Active** in Oracle Access Governance. From licensing and pricing viewpoint, we only bill unique **Active** identities in Oracle Access Governance. Further, Active identities can be flagged either as *Workforce* or *Consumer* identities by the Oracle Access Governance administrator. The main difference is that a *Consumer* user cannot log on to Oracle Access Governance but you can manage permissions or provision these identities with a fixed set of privileges using Oracle Access Governance.

For example, in a financial institution, employees, such as accountants, tellers, managers, or administrative staff can be your *Workforce* user, whereas bank account owners, insurance policy holders, and others customers can be your *Consumer* identities. For more information, see Manage Identities.

Based on these identity types, we have defined our unit metrics:

- **Workforce user per month**: A unique identity that is configured to access the service either through a user interface or through programmatic configuration during the billing period, regardless of whether the identity is actively accessing the service at any given time. Workforce user can be an individual, such as an employee or contractor, or a service identity, such as bots,applications, or services.
- **Consumer user per month**: A unique identity that is not configured to access the service through either a user interface or through a programmatic configuration during the billing period, but whose accesses are managed in the Oracle Access Governance Console by workforce identities. Consumer identities can be an individual, such as customers, alumni, outsourced partners, or devices.

So, the metric for Oracle Access Governance Stock Keeping Units (SKUs) is *per month*. For billing, only the **Active** identities on an hourly basis are considered and we will generate the bill for the entire month. If you don’t mark any identity as **Active**, you will not be billed for Oracle Access Governance.

User Segments or Tiers within each SKU

Based on the type of licensing models and identity types, Oracle Access Governance extends discounts by offering multiple tiers based on number of workforce identities. See the tier details in the Licensing Plan for Oracle Access Governance Licensing Plan for Oracle Access Governance.

Licensing Models and Offerings

Here's a detailed plan on licensing models and offerings for Oracle Access Governance.

Table 1-1 Licensing Plan for Oracle Access Governance

Licensing Models	Identity Types	Tiers
Oracle Access Governance for Oracle Cloud Infrastructure	Workforce Identity	• First 100,000 workforce identities • More than 100,000 workforce identities
Oracle Access Governance for Oracle Workloads	Workforce Identity	• First 10,000 workforce identities • More than 10,000 workforce identities and up to 30,000 workforce identities • More than 30,000 workforce identities
Oracle Access Governance for Oracle Workloads	Consumer Identity	No Tiers
Oracle Access Governance Premium	Workforce Identity	• First 10,000 workforce identities • More than 10,000 and up to 30,000 workforce identities • More than 30,000 workforce identities
Oracle Access Governance Premium	Consumer Identity	No Tiers

Choosing the Right Licensing Model

Let's consider a few scenario examples that will help you select the right licensing model based on your requirement.

Oracle Access Governance Licensing for Users beyond Oracle Applications

Let's say you want to manage governance for a total 1000 identities, breakdown as follows:

- Employee data of 800 identities available in **Oracle Identity Governance (OIG)**
- Contractors data of 200 identities available in **Microsoft Entra ID**

In such a case to manage governance of employees and contractors data, you need all applications and services, so you need subscription to **Oracle Access Governance Premium**. You will get integration support for OCI IAM, Oracle workloads and all other available applications and services. You will be billed for identities marked as **Active** in Oracle Access Governance.

Oracle Access Governance Licensing to Manage Domain Identities in Oracle Cloud Infrastructure

Let's say you want to manage governance for three (3) identity domains, having 1000 unique identities in each domain, making a total of 3000 identities. As you only need to manage OCI IAM domains, you would need subscription to **Oracle Access Governance for Oracle Cloud Infrastructure**. If you want to manage governance for all the 3000 identities, you first need to mark these identities as *Active* in Oracle Access Governance, and you will be billed for these active identities.

This article provides details of suggested preparatory steps you can complete before starting with Oracle Access Governance.

Oracle Cloud Infrastructure Services

When you order Oracle Access Governance through Universal Credits, you automatically get access to Oracle Cloud Infrastructure and other required services.

Here's some information about how Oracle Access Governance uses other services and what you need to do if you're setting up Oracle Access Governance for the first time.

Table 1-2 Oracle Cloud Infrastructure Services used by Oracle Access Governance

Service	What is it for?	Do I need to do anything?
Oracle Cloud Infrastructure Identity and Access Management	**Compartments**: You use compartments to organize resources on Oracle Cloud Infrastructure. **Policies**: You use IAM security policies to grant permissions. **Domains**: You use identity domains to manage users and groups in your organization who will be required to use Oracle Access Governance and Oracle Cloud Infrastructure Console.	Yes Before you create your first Oracle Access Governance instance, Oracle recommends that you set up one or more compartments in which you can deploy and secure your cloud resources. • Setting Up Your Tenancy • Managing Compartments Optionally, you can set up security policies that give other users permission to set up and manage Oracle Access Governanceinstances. See Set Up Users for further details.
Oracle Identity Cloud Service	If identity domains aren't available in your cloud account, you use Oracle Identity Cloud Service to manage the users and groups in your organization who will use Oracle Access Governance. In most cases, Oracle Access Governance is automatically federated with the *primary* Oracle Identity Cloud Service instance associated with your tenancy.	Yes You can add users and groups before you create the Oracle Access Governance instance or after; you can decide. See Set Up Users for further details. If you want to federate with a secondary Oracle Identity Cloud Service instance or your tenancy is a government region where federation isn't set up automatically, you must federate with Oracle Identity Cloud Service manually.

Typical Workflow for Administrators

This topic outlines a typical workflow for Oracle Access Governance administrators.

If you’re setting up Oracle Access Governance for the first time, follow these tasks as a guide.

Task	Description	More Information
Place an order for Oracle Access Governance or sign up for a free Oracle Cloud promotion	Sign up for a free credit promotion or subscribe to Oracle Access Governance through Universal Credits. See Oracle Global Infrastructure Regions	Request and Manage Free Oracle Cloud Promotions Upgrade Your Free Oracle Cloud Promotion
Activate your Oracle Cloud account and sign in for the first time	You receive a welcome email when your account is ready. To activate your account, you must sign in with the credentials provided in the email. As the Cloud Account Administrator, you can complete all the setup tasks for Oracle Access Governance.	Manage Service Instance
Determine your service requirements	Plan your Oracle Access Governance deployment. Think about what you need before you start.	Service Requirements • Users • Region
(Optional) Enable other users to set up services	If you don’t want to set up Oracle Access Governance yourself, give other users permissions to create services.	Set Up Users
(Recommended) Create a compartment for your service	Create a compartment for your Oracle Access Governance deployment.	When you sign up for Oracle Cloud Infrastructure, Oracle creates your tenancy with a root compartment that holds all your cloud resources. You then create additional compartments within the tenancy (root compartment) and corresponding policies to control access to the resources in each compartment. Before you create an Oracle Access Governance instance, Oracle recommends that you set up the compartment where you want the instance to belong. You create compartments in Oracle Cloud Infrastructure Identity and Access Management. See Setting Up Your Tenancy and Managing Compartments.
Create a service instance	Deploy a new service with Oracle Access Governance.	Create Service Instance
Verify your service instance	When your service is ready, check that you can sign in and your service is up and running.	Verify Service Instance
Set up users and groups	Set up users and groups for Oracle Access Governance and assign them to application roles.	Set Up Users

Task	Description	More Information
Integrate with Authoritative Sources	Configure integration with external systems from which you eish to onboard identities.	See: • Identity Orchestration Overview • Oracle Access Governance Integrations
Activate Identities for License Management	Define which identities can use your Oracle Access Governance service.	Activate Identities for License Management
Integrate with Managed Systems	Configure integration with external systems which you want to perform access reviews and campaigns against	See: • Identity Orchestration Overview • Oracle Access Governance Integrations
Administer services	Monitor services and perform administrative tasks such as edit and delete. Delegate administrative responsibilities to others through security policies.	See Manage Service Instance and Set Up Users.

You can set up user accounts for everyone you expect to use Oracle Access Governance before or after you create your Oracle Access Governance instance.

About Setting Up Users and Groups

Set up user accounts for everyone you expect to use Oracle Access Governance.

The way you manage users for Oracle Access Governance (and Oracle Cloud Infrastructure) depends on whether *identity domains* are available in your cloud account.

- **Oracle Cloud Infrastructure Identity and Access Management (IAM) Identity Domains**: Some Oracle Cloud regions have been updated to use identity domains. If you have a new cloud account in one of these regions, you use identity domains to manage the users who perform tasks in both Oracle Access Governance and Oracle Cloud Infrastructure.
- **Oracle Identity Cloud Service**: If you have an existing cloud account or you deploy Oracle Access Governance in a region that does not currently offer identity domains, you use a federated Oracle Identity Cloud Service to manage the users who perform tasks in Oracle Access Governance. In addition, you use Oracle Cloud Infrastructure Identity and Access Management to manage the users who create and manage your Oracle Access Governance deployments using the Oracle Cloud Infrastructure Console.

It is easy to determine whether or not your cloud account offers identity domains. In Oracle Cloud Infrastructure Console, navigate to **Identity & Security**. Under **Identity**, check for **Domains**.

- If you see **Domains**, you use identity domains to manage users and groups for Oracle Cloud Infrastructure and your Oracle Access Governance deployments. See Use Identity Domains to Onboard Users and Groups for Oracle Access Governance.

- If **Domains** is not listed, you use a federated Oracle Identity Cloud Service to manage Oracle Access Governance users and IAM to manage Oracle Cloud Infrastructure users. See Use Oracle Identity Cloud Service to Onboard Users and Groups for Oracle Access Governance.

The following table outlines the differences between the two configurations.

Cloud Accounts That Use Identity Domains	Cloud Accounts That Don't Use Identity Domains
Users and groups are configured in IAM.	Users and groups are configured in both IAM and Oracle Identity Cloud Service, and are linked through federation.
Provides a single, unified console for managing users, groups, dynamic groups, and applications in *domains*.	Oracle Cloud Infrastructure Identity and Access Management must be federated with Oracle Identity Cloud Service.
Provides Single Sign-On to more applications using a single set of credentials and a unified authentication process.	Requires separate federated credentials for Oracle Identity Cloud Service.
The Federation page doesn't list any entries for Oracle Identity Cloud Service.	The Federation page lists **oracleidenitycloudservice**, the primordial Oracle Identity Cloud Service automatically federated in your cloud account.

Use Identity Domains to Onboard Users and Groups for Oracle Access Governance

If your Oracle Access Governance instance uses *identity domains* for identity management, you use Oracle Identity Governance provisioning, an external Identity Provider (IDP), or a self-registration profile to onboard user accounts for everyone you expect to use Oracle Access Governance. These users will be assigned to a group, which, when you have completed onboarding, you map to Oracle Access Governance *application roles*)

As an Oracle Cloud Infrastructure *Cloud Administrator*, you can use one of the following approaches to enable access for users to the Oracle Access Governance application.

Approach 1: Set Federated Authentication from an External Identity Provider (IDP)

1. Setup federation with an external IDP:
   a. Set up a federated login between an Identity Domain and external IDP. Users can sign in and access Oracle Access Governance resources and features by using existing logins and passwords managed by the IDP.
   b. Refer to Managing Identity Providers in the Oracle Cloud Infrastructure documentation for further details.
2. Enable SAML Just-In-Time provisioning.
   a. This process automates user account creation when a user first tries to sign in to Oracle Cloud Infrastructure where the user does not yet exist in the Identity Domain.
   b. Refer to About SAML Just-In-Time Provisioning in the Oracle Cloud Infrastructure documentation for further details.

Approach 2: Configure Oracle Identity Governance Provisioning with Oracle Cloud Infrastructure Identity and Access Management Using the Oracle Identity Cloud Service Application

1. Configure the Oracle Identity Cloud Service Application.
    a. Download the connector installation package and copy the contents to the *OIG_HOME/server/ConnectorDefaultDirectory* directory. Refer to Downloading the Connector Installation Package for further details.
    b. Log in to the Oracle Cloud Infrastructure Console and create an application with the type *Confidential*. Refer to Creating an Application By Using the Connector for further details.
    c. Copy the *Client ID* and *Client Secret* from the created Application. This will be used in customAuthHeaders in ITResource.
    d. Configure SSL to secure communication between Oracle Identity Governance and the target system, in this case, Oracle Access Governance. Refer to Configuring SSL for the Connector for further details.
2. Create Groups: Login to the Oracle Cloud Infrastructure Console and create groups for any Oracle Identity Governance groups you want to map to Oracle Access Governance roles.
3. Create an IDCS application in Oracle Identity Governance. Refer to Creating an Application By Using the Connector for further details.
4. Run the Group Lookup Recon Job.
5. Provision the IDCS application for those users with a membership of Access Governance groups.

Approach 3: Self Registration Profiles

Create self-registration profiles to enable users to create their accounts in Oracle Cloud Infrastructure Identity and Access Management. Refer to Creating Self-Registration Profiles for further details.

Use Oracle Identity Cloud Service to Onboard Users and Groups for Oracle Access Governance

If your Oracle Access Governance instance uses Oracle Identity Cloud Service for identity management, you use Oracle Identity Governance provisioning, an external Identity Provider (IDP), or a self-registration profile to onboard user accounts for everyone you expect to use Oracle Access Governance. These users will be assigned to a group, which, when you have completed onboarding, you map to Oracle Access Governance *application roles*.

As an Oracle Cloud Infrastructure *Cloud Administrator*, you can use one of the following approaches to enable access for users to the Oracle Access Governance application.

Approach 1: Set Federated Authentication from an External Identity Provider (IDP)

1. Setup federation with an external IDP:
    a. Set up a federated login between an Identity Domain and external IDP. Users can sign in and access Oracle Access Governance resources and features by using existing logins and passwords managed by the IDP.
    b. Refer to Federating with Identity Providers in the Oracle Cloud Infrastructure documentation for further details.

2. Enable SAML Just-In-Time provisioning.
    a. This process automates user account creation when a user first tries to sign in to Oracle Cloud Infrastructure where the user does not yet exist in the Identity Domain.
    b. Refer to User Provisioning for Federated Users in the Oracle Cloud Infrastructure documentation for further details.

Approach 2: Configure Oracle Identity Governance Provisioning with Oracle Cloud Infrastructure Identity and Access Management Using the Oracle Identity Cloud Service Application

1. Configure the Oracle Identity Cloud Service Application.
    a. Download the connector installation package and copy the contents to the *OIG_HOME/server/ConnectorDefaultDirectory* directory. Refer to Downloading the Connector Installation Package for further details.
    b. Log in to the Oracle Cloud Infrastructure Console and create an application with the type *Confidential*. Refer to Creating an Application By Using the Connector for further details.
    c. Copy the *Client ID* and *Client Secret* from the created Application. This will be used in customAuthHeaders in ITResource.
    d. Configure SSL to secure communication between Oracle Identity Governance and the target system, in this case, Oracle Access Governance. Refer to Configuring SSL for the Connector for further details.
2. Create Groups: Login to the Oracle Cloud Infrastructure Console and create groups for any Oracle Identity Governance groups you want to map to Oracle Access Governance roles.
3. Create an IDCS application in Oracle Identity Governance. Refer to Creating an Application By Using the Connector for further details.
4. Run the Group Lookup Recon Job.
5. Provision the IDCS application for those users with a membership of Access Governance groups.

Approach 3: Self Registration Profiles

Create self-registration profiles to enable users to create their accounts in Oracle Cloud Infrastructure Identity and Access Management. Refer to Creating Self-Registration Profiles for further details.

Assign Access Governance Application Roles to Users and Groups

Once users are on-boarded, all active Workforce users can log in and access the Oracle Access Governance Console. To determine what privileges they have within Oracle Access Governance, assign them the relevant predefined application roles as described in Predefined Application Roles Reference.
Assigning Oracle Access Governance depends on whether identity domains are available in your cloud account.

- If **Domains** is listed, then use the For Identity Domain Users method.
- If **Domains** is not listed, and you use the federated Oracle Identity Cloud Service (IDCS) method to manage users, then use the For Non Identity Domain Usersmethod.

For Identity Domain Users

Here's how you can assign Oracle Access Governance application roles to Users and Groups:

1. Open your web browser and navigate to https://cloud.oracle.com.
2. Enter the name of your **Cloud Account Administrator** in the **Cloud Account Name** field and click **Next**.
3. On the Cloud Infrastructure sign-in page, enter your sign-in credentials under **Oracle Cloud Infrastructure Direct Sign-In**. Click **Sign In**.
4. Click the  icon in the top, left corner to display the navigation menu.
5. Click **Identity & Security** in the navigation menu.
6. Select **Domains** within the **Identity** list.
7. On the left pane, in the **Compartment** list, select the relevant compartment for Oracle Access Governance.
8. In the available domain list, select the domain link related to Oracle Access Governance. Your selected domain page is displayed.
9. From the left pane, select the **Oracle Cloud Services** tab.
10. Select the Oracle Access Governance cloud service.
11. On the left pane, in the **Resources** section, select **Application roles**.
12. In the **Application roles** section, select the icon corresponding to the application role that you want to assign.
13. Select the **Manage** link corresponding to the **Assigned users** category. The **Manage user assignments** window is displayed.

> **Note:**
>
> To assign application roles to user groups, select the **Manage** link corresponding to the **Assigned groups** category.

14. Select the **Show available users** link.
15. In the available list of users, select the check box corresponding to the user name, and then click **Assign**.

The application role is assigned to the selected user or group. You can verify the same by viewing the names in the **Available users** or **Available groups** list.

For Non Identity Domain Users

Here's how you can assign Oracle Access Governance application roles to Users and Groups for Non Identity Domain-based users:

- Sign in to the Oracle Identity Cloud Service console with the user assigned as the *Service Administrator* team role.
- Click the  icon in the top, left corner to display the navigation menu.
- Click **Oracle Cloud Services** and then select your Oracle Access Governance service instance.

- Click the **Application Roles** tab. All the available application roles in Oracle Access Governance are displayed.
- Click the  role menu icon corresponding to the application role that you want to assign, and then, as per your requirement, select **Assign Users** or **Assign Groups**.

Application Roles

About Application Roles in Oracle Access Governance

Oracle Access Governance offers several predefined application roles with different capabilities levels to perform the access management and governance operations. You can assign one or more application roles to users from your Oracle Access Governance cloud service instance. You can't modify predefined application roles or modify permissions assigned within these roles.

Administrator (`AG_Administrator`)

Oracle Access Governance Administrator has the highest level of access within Oracle Access Governance. Users with the Administrator role are responsible for managing all Oracle Access Governance operations, including managing Orchestrated systems, access controls, service administrative operations, and so on.

The prime responsibility of an Administrator is to

- Define foundational tasks available as part of the Service Administration module in Oracle Access Governance, such as setting up Orchestrated systems, managing identities, configuring core and custom identity attributes, configuring notifications, verifying data load operations in Oracle Access Governance.
- Configure Event-based Access Reviews to perform micro-certifications and manage unmatched accounts.

For example, you can assign `AG_Administrator` to Security Administrators or Identity and Access Management Specialist to manage your Oracle Access Governance cloud service instance.

Typically, `AG_Administrator` would establish the first integration with the Authoritative source by creating an Orchestrated System, executing the full data load, setting rules to define *Workforce* and *Consumer* users. `AG_Administrator` can then assign owners to manage the Orchestrated system to any Oracle Access Governance active user.

An `AG_Administrator` has the full access to all features and functionalists within the service instance. They have the all the permissions to create, view, update, and delete the Oracle Access Governance resources:

- Orchestrated System
- Identity Collections
- Access Bundles
- Roles
- Policies
- Approval Workflows

Service Desk Administrator (AG_ServiceDesk_Admin)

Oracle Access Governance Service Desk Administrator is responsible for performing advanced account administrative functions directly within Oracle Access Governance. The prime responsibility of a Service Desk Administrator is to perform highly critical and urgent operations without the need of any approvals, especially related to Account Lifecycle Management operations.

Users with the Service Desk Administrator role can perform the account administrative functions from the **Service Administration** , and then **Manage Identities**, and then **Identities** page:

- View identity details for all the identities.
- View account details for permissions.
- Terminate all the accounts and accesses for an identity at once without any approvals. Once terminated, you can re-provision or activate the accounts and accesses, with Grant Type *Policy*.
- Enable, Disable or Delete one or multiple accounts for an identity.
- Revoke one or more permissions assigned directly from the Managed System or provisioned through request.
- Retry provisioning for failed or pending status.
- Change Password for an account managed by Oracle Access Governance.
- Manage Delegations for approvals or access reviews.

For example, you can assign `AG_ServiceDesk_Admin` to an IT Specialist to immediately terminate all accounts and accesses based on an incident response triggered by repeated failed login attempts to prevent potential unauthorized activity.

Additionally, `AG_ServiceDesk_Admin` can perform the following operations as part of Oracle Access Governance user:

- View orchestrated systems details along with activity logs.
- As a resource owner, view, update, or delete Oracle Access Governance resources that they own.
- As a reviewer associated with Approval workflows, approve access requests and review access tasks.
- As a user, can view the assigned privileges for self and direct reports.
- As an access reviewer, can review and certify the access review tasks if associated with a specific approval workflow.
- Manage Delegations

Campaign Administrator (AG_CampaignAdmin)

Oracle Access Governance Campaign Administrators can initiate an access review process by creating Campaigns. They can modify, delete, and monitor self-created access review campaigns. They can view campaign report and download CSV data for offiline purposes.

Their prime responsibility is to schedule ad-hoc or periodic campaigns for Identity Access Reviews, Policy Reviews, Identity Collection Reviews, or Resource Ownership review across all systems.

Additionally, the Campaign Administrators:

- Can create approval workflows
- Can create identity collections
- As a resource owner modify, delete, and view resources that they own
- As a reviewer associated with Approval workflows, approve access requests and review access tasks.
- As a user, can view the assigned privileges for self and direct reports.

Enterprise-wide Browser Access Administrator (`AG_Enterprise_Wide_Access_Admin`)

Oracle Access Governance Enterprise-wide Access Administrator get the comprehensive visibility on all the components, access information, and resources within an enterprise framework from the **Who Has Access to What → Enterprise-wide Browser** page.

Primarily, Enterprise-wide Access Administrators can:

- Browse access information using various perspectives, such as Identities, Identity Collections, Roles, Permissions, Policies, Resources, and Organizations.
- Run User-created reviews for identities, identity collections, policies from the Enterprise-wide Browser dashboard.
- Generate a monthly report on access reviews created from Enterprise-wide Browser.
- Download CSV and PDF screenshot.

Additionally, can:

- Can create Identity Collections
- As a resource owner, view, update, or delete Oracle Access Governance resources that they own.
- As a reviewer associated with Approval workflows, approve access requests and review access tasks.
- As a user, can view the assigned privileges for self and direct reports.

Application Owner Administrator (`AG_AppOwner_Admin`)

Oracle Access Governance Application Owner Administrator is responsible for performing integrations with other systems by adding an Orchestrated system, modifying the connection settings, validating and loading the data in Oracle Access Governance. They can also configure an orchestrated system by editing the integration settings, configuring notification settings, defining transformation rules for inbound and outbound data for identity and account attributes, and defining correlation rules for matching identities and identity accounts.

Application Owner Administrator is primarily responsible to:

- Set up integrations with an application as an Authoritative Source or a Managed System by creating an orchestrated system.
- Manage and configure the integrated systems.

> **Note:**
>
> `AG_AppOwner_Admin` cannot activate identities or configure identity attributes for an orchestrated system. To do so, you need the `AG_Administrator` role.

Additionally, Application Owner Administrator:

- Can create approval workflows
- Can create identity collections
- Can create access bundles
- As a resource owner, can modify, delete, and view resources that they own. Resources can be Access Bundles, Identity Collections, Policies, Approval Workflows, Orchestrated Systems, Organizations, and Roles.
- As a reviewer associated with Approval Workflows, can approve access requests and review access tasks.
- As a user, can view the assigned privileges for self and direct reports.
- As a user, can use the self-service module to request new access, track requests, assign preferences, and so on.

Application Owner Restricted Administrator (`AG_AppOwner_Admin_Restricted`)

Oracle Access Governance Application Owner Restricted Administrator is responsible for creating a new integration with other systems by adding an orchestrated system. However, they can manage integrations and configure settings only for systems that they own as a resource owner.

Application Owner Restricted Administrator is primarily responsible to:

- Set up integrations with an application as an Authoritative Source or a Managed System by creating an orchestrated system.
- Manage and configure the orchestrated system for which it is the resource owner.

> **Note:**
>
> `AG_AppOwner_Admin_Restricted` cannot activate identities or configure identity attributes for an orchestrated system. To do so, you need the `AG_Administrator` role.

Additionally, Application Owner Restricted Administrator:

- Can create approval workflows
- Can create identity collections
- Can create access bundles
- As a resource owner modify, delete, and view resources that they own. Resources can be Access Bundles, Identity Collections, Policies, Approval Workflows, Orchestrated systems, Organizations, and Roles.
- As a reviewer associated with approval workflows, approve access requests and review access tasks.
- As a user, can view the assigned privileges for self and direct reports.

- As a user, can use the self-service module to request new access, track requests, assign preferences, and so on.

Scenario: If *Betty* is assigned the `AG_AppOwner_Admin_Restricted` role, Betty can establish new integrations by creating a new orchestrated system from the **Service Administration** → **Orchestrated Systems** page. However, Betty can manage and configure settings for the orchestrated systems only if *Betty* is assigned as the resource owner (primary owner or one of the additional owners) for the orchestrated system resource.

Access Control Administrator (`AG_AccessControl_Admin`)

Oracle Access Governance Access Control Administrator is responsible for managing Access Control Administration in Oracle Access Governance.

Access Control Administrator is primarily responsible to:

- Create and Manage Identity Collections
- Create and Manage Access Bundles
- Create and Manage Approval Workflows
- Create and Manage Roles
- Create and Manage Policies
- Create and Manage Organizations from the **Manage Identities** page

Additionally, Access Control Administrator:

- As a resource owner, can modify, delete, and view resources that they own. Resources can be Access Bundles, Organizations, Identity Collections, Policies, Approval Workflows, Orchestrated systems, and Roles.
- As a reviewer associated with Approval workflows, can approve access requests and review access tasks.
- As a user, can view the assigned privileges for self and direct reports.
- As a user, can use the self-service module to request new access, track requests, assign preferences, and so on.

Access Control Restricted Administrator (`AG_AccessControl_Admin_Restricted`)

Oracle Access Governance Access Control Restricted Administrator is responsible for creating Access Controls resources in Oracle Access Governance.

Access Control Restricted Administrator is primarily responsible to:

- Create Identity Collections, Access Bundles, Approval Workflows, Roles, Policies, and Organizations.

Additionally, Access Control Restricted Administrator:

- As a resource owner, can modify, delete, and view resources that they own. Resources can be Access Bundles, Organizations, Identity Collections, Policies, Approval workflows, Orchestrated systems, and Roles.
- As a reviewer associated with Approval workflows, can approve access requests and review access tasks.
- As a user, can view the assigned privileges for self and direct reports.
- As a user, can use the self-service module to request new access, track requests, assign preferences, and so on.

Scenario: If *Betty* is assigned the `AG_AccessControl_Admin_Restricted` role, Betty can implement access control specific resources by creating new identity collections, approval workflows, policies, roles, package permissions into Access Bundles using the **Access Controls** module. However, Betty can manage these resources only if *Betty* is assigned as the resource owner (Primary Owner or one of the Additional Owners) for the resources.

Auditor (`AG_AUDITOR`)

Oracle Access Governance Auditor role is responsible for monitoring all the campaigns. They can view campaign details, download access review report for each campaign. In addition to viewing report, Auditor can save the reports offline in PDF format or download the CSV data for record-keeping or further analysis or audit.

Additionally, depending on the ownership provided within Oracle Access Governance, an auditor can:

- As a campaign owner, can modify, delete, monitor self-owned access review campaigns.
- As an access reviewer, can review and certify the access review tasks if associated with a specific approval workflow.
- As a resource owner, view, modify and delete resources that they own.
- Create Identity Collections.
- Manage Identity Collections that they own.

User (`AG_USER`)

Oracle Access Governance user is an end user responsible for viewing and managing their accesses using Oracle Access Governance. All the Oracle Access Governance Active Workforce users are assigned this role, by default.

Your cloud domain administrator can also assign this application role (`AG_USER`) from the OCI cloud service page. Users primarily engage in self-service tasks, which can include requesting permissions through Access Bundles or Roles, viewing access details, managing preferences, changing account passwords, and so on.

Additionally, depending on the ownership provided within Oracle Access Governance, an end user:

- As a campaign owner, can modify, delete, monitor self-owned access review campaigns.
- As an access reviewer, can review and certify the access review tasks if associated with a specific approval workflow.
- As a resource owner, view, modify and delete resources that they own.
- Create Identity Collections.
- Manage Identity Collections that they own.

Application Roles and Responsibilities Reference

Lists all predefined application roles and corresponding responsibilities. Assign users one or more predefined Oracle Access Governance application roles to start your Identity Governance and Administration journey with Oracle Access Governance .

Predefined Application Roles

Oracle Access Governance offers several predefined roles to get you started. A single user can hold multiple application roles, as needed.

Here's a list of application roles in Oracle Access Governance:

Table 1-3 Oracle Access Governance Application Roles

Application Role	Access and Action
Administrator `AG_Administrator` See Details	• **Orchestrated Systems**: Create and Manage integrations of Authoritative Sources and Managed systems applications with Oracle Access Governance as Orchestrated systems. • **Access Controls**: – Create and Manage Roles – Create and Manage Identity Collections – Create and Manage Organizations – Create and Manage Policies – Create and Manage Approval Workflows • **Access Reviews**: – Create Campaigns. – Modify, Delete, Monitor all access review campaigns. – Enable and Disable Event-based access reviews for micro-certifications. – Modify, Delete, Monitor all event-based access reviews – Define auto-action for low-risk access reviews or unmatched accounts – Generate Event-Based Access Report and Access Reviews Campaign Report • Manage Identities and Identity Attributes • Deactivate or Activate all accounts and associated accesses for an identity managed by Oracle Access Governance • Revoke one or more permissions assigned directly from the Managed System or provisioned through request. • Retry Provisioning for Failed or Pending Status • Enable, Disable, Delete one or multiple accounts • Manage Notifications • **Settings**: – Create, Modify Security Settings – Create, Modify Systems Settings • **Delegations**: – Create, Modify own delegations – Create, Modify other user's delegations • **Who Has Access to What - Enterprise-wide Browser** – View Enterprise-wide Access Insights – Create user-created access reviews and download related reports – Download CSV reports and PDF screenshot

Table 1-3 (Cont.) Oracle Access Governance Application Roles

Application Role	Access and Action
Service Desk Administrator `AG_ServiceDesk_Admin` See Details	Related to advanced account administrative functions performed directly from the **Service Administration** → **Manage Identities** page. • View Identity details • Terminate or Activate all accounts and accesses for an identity managed by Oracle Access Governance • Enable, Disable, Delete accounts • Retry Provisioning for Failed or Pending Status • Revoke permissions assigned directly from the Managed System or provisioned through request. • Manage delegations • Change Password • View a list of orchestrated system defined in the Oracle Access Governance service instance.
Campaign Administrator `AG_CampaignAdmin` See Details	Related to **Access Reviews** • Create Campaigns • Modify, Delete, Monitor self-created campaigns
Enterprise-wide Access Administrator `(AG_Enterprise_Wide_Access_Admin)` See Details	Related to **Who Has Access to What - Enterprise-wide Browser** • View access insights across an enterprise using **Enterprise-wide Browser** • Create user-created access reviews and view corresponding time-range-based access review reports • Download CSV reports and PDF screenshot
Access Control Administrator `AG_AccessControl_Admin` See Details	Related to **Access Controls** • Create and Manage Roles • Create and Manage Identity Collections • Create and Manage Policies • Create and Manage Approval Workflows • Create and Manage Access Bundles • Create and Manage Organizations
Access Control Restricted Administrator `AG_AccessControl_Admin_Restricted` See Details	Related to **Access Controls** • Create Roles, Identity Collections, Policies, Approval Workflows, Access Bundles, Organizations • Manage resources that they own, as a resource owner
Application Owner Administrator `AG_AppOwner_Admin` See Details	Related to **Orchestrated Systems** • Create Orchestrated systems to perform new integrations. • Manage and configure all Orchestrated system defined in the Access Governance service instance. Related to **Access Controls** • Create Access Bundles, Approval Workflows and Identity Collections • Manage resources that they own, as a resource owner

Table 1-3 (Cont.) Oracle Access Governance Application Roles

Application Role	Access and Action
Application Owner Restricted Administrator `AG_AppOwner_Admin_Restricted` See Details	Related to **Orchestrated Systems** • Create Orchestrated systems to perform new integrations. • Manage and Configure Orchestrated system settings that they own, as resource owner. Related to **Access Controls** • Create Access Bundles, Approval Workflows and Identity Collections. • Manage resources that they own, as a resource owner.
Auditor `AG_Auditor` See Details	Related to **Access Reviews** • Monitor all access review campaigns
User `AG_User` See Details	• As a campaign owner - modify, delete, monitor self-owned access review campaigns. • As an access reviewer - review and certify the access review tasks if associated with a specific approval workflow. • As an end user - manage the self-service module to view your own accesses, change account password, request access, set preferences, set delegations, manage approvals or track access requests for self or direct reports. • As a resource owner, view, modify and delete resources that they own. • Create Identity Collections **Note:** All Oracle Access Governance active Workforce users are assigned the `AG_User` by default. All the active Workforce users can log on to Oracle Access Governance Console.

2

What's New

What's new in Oracle Access Governance

Here’s an overview of new features and enhancements added recently to improve your Oracle Access Governance experience.

Recent Updates in Oracle Access Governance

Here’s an overview of new features released, including documentation updates.

March 2025 Update

Oracle Access Governance REST APIs Availability

Feature	Description
Oracle Access Governance REST APIs Availability	Oracle Access Governance released REST APIs to automate and extend Identity Governance and Administration (IGA) capabilities. The REST APIs will be available with an **Oracle Access Governance Premium** license. You can streamline access control processes, improve compliance, and enhance the overall identity management across diverse environments. For more details, see Oracle Access Governance REST APIs.

Audit Events and Event Data Publisher in Oracle Access Governance

Feature	Description
Audit Events as part of Event Data Publisher	You can now get comprehensive audit trail of actions performed within Oracle Access Governance, in near real-time for compliance, creating customized reports, or for troubleshooting purposes. To configure receiving audit events in your OCI Stream, select the **Audit event** option on the **Data Feed** configuration page. You'll receive a JSON response, containing detailed interactions points, such as who initiated the request, which resource IDs were involved, what action was performed, request and response payloads for each action, along with additional details.
Additional Oracle Access Governance components supported for Event Data Publisher	You can now receive additional data events, such as Access Guardrails, Permission Assignments, Roles, and so on for all Orchestrated systems managed by Oracle Access Governance. The setup configuration remains unchanged, Day 0 events will be exported to OCI Buckets and subsequent events are published to OCI Streams.

Integrate Oracle Access Governance with Orchestrated Systems

Feature	Description
Integrate with	: Oracle Access Governance now supports identity orchestration for provisioning and reconciliation of accounts in **Atlassian JIRA** as a Managed System. is a comprehensive project management and issue tracking tool that enables teams to plan, prioritize, monitor tasks effectively. With this integration, you can create and manage identity accounts and group assignment from Oracle Access Governance.
Integrate with Human Capital Management (HCM)	For , you can now: • Perform data load of identities for an employee or contractor data not associated with any user profile in . • Use extended support of default attribute to manage accounts. To support these feature updates, the existing configuration remains unchanged. However, as a prerequisite, an extended setup to create database views, synonyms, and grant permissions are required.
Integrate with Oracle Fusion Cloud Applications Human Capital (HCM)	For Oracle Fusion Cloud Applications Human Capital (HCM): • **Authoritative Source:** You can now perform identity data load and manage accounts based on a **Person** record, even if there is no associated **User Account** for that identity. • **Managed System:** You can create and manage SCIM user accounts based on the Person record managed by Oracle Access Governance. For successful provisioning, you must add an outbound transformation rule to set the Person Number attribute value.

Managed System Account Management - Account Lifecycle Settings

Feature	Description
Account Lifecycle Settings	In the Orchestrated System **Account Lifecycle** settings, the following enhancements have been done: • For a Joiner use case, you can now configure whether Oracle Access Governance should create new accounts or manage permissions only for the reconciled accounts. If the **Allow Access Governance to create new accounts** option is unchecked, Oracle Access Governance will not create new accounts but can only manage permissions for the existing accounts. • For a Leaver use case, select **Include accounts that are not created by Access Governance** to disable or delete direct accounts (`grant_type Direct`) not provisioned or managed through Oracle Access Governance. • For accounts with no remaining permissions (mover or leaver), you can now choose to **Delete**, **Disable**, or take **No action**. If you select **No action**, accounts will not be disabled or revoked and the user may have access to the accounts beyond the Oracle Access Governance scope.

Additional Service Enhancements

Feature	Description
Campaigns → My Access Reviews	You can now search identity access reviews based on **Oracle Access Governance** Organization name.

Feature	Description
Who Has Access to What → Enterprise-wide Browser Compartment Reports	For an OCI Orchestrated system, you can now you can now export compartment report using the **Export compartment report** button from the **Enterprise-wide Browser** page to view available resource details in that compartment along with access information for identities associated with these resources.

February 2025 Update

Integrate Oracle Access Governance with New Orchestrated Systems

Feature	Description
Integrate with	: Oracle Access Governance now supports onboarding of identities, provisioning, and reconciliation of accounts in as an Authoritative Source and as a Managed System. is a cloud-based Human Capital Management (HCM) software that helps organizations manage various HR functions, including talent management, onboarding, payrolls, or other HR processes. With this integration, you can manage identity accounts and group assignment from Oracle Access Governance
Integrate with	: Oracle Access Governance now supports onboarding of identities, provisioning, and reconciliation of accounts in HCM as an Authoritative Source and as a Managed System. is a cloud-based Human Capital Management (HCM) and workforce management software that helps organizations streamline various HR functions, including talent management, workforce planning, onboarding, payrolls, or other core HR processes. With this integration, you can manage identity accounts and security group assignment from Oracle Access Governance.

Modify Account Attributes from Oracle Access Governance Console

Feature	Description
Modify Account Attributes	As a Service Desk Administrator (`AG_ServiceDesk_Admin`), Oracle Access Governance allows you to directly update default or custom account attributes without any approval workflow. From the **Manage Identities**, and then **Account details** page. Click the **Edit Account** operation to modify the value of account attributes. After you have updated the account attributes, it triggers the **Update Account** operation on the Orchestrated system. For example, based on the attributes supported for an Orchestrated system, you can use this feature to update Account Name, Locked status, address change, password and so on.

January 2025 Update

Access Guardrails in Oracle Access Governance

Feature	Description
Access Guardrails in Oracle Access Governance	**Access Guardrails** in Oracle Access Governance, allows you to establish preventive access control measures to ensure that authorized and compliant identities gain access. You can define a set of conditions that an identity must meet before gaining an access to a permission — such as completing mandatory trainings, or meeting policy requirements. If these conditions are not met, a violation is raised, and you can choose to block access immediately or allow a grace period for compliance.

AI-Powered Access Bundle Recommendation Engine in Oracle Access Governance

Feature	Description
AI-Powered Access Bundle Recommendation Engine in Oracle Access Governance	Oracle Access Governance now supports AI-powered intelligent Access Bundle Recommendation capability to instantly get a list of pre-bundled sets of permissions, based on usage patterns and relationship mapping from Managed Systems. Instead of manually creating access bundles and associating permissions, the system suggests access bundles that you can accept, edit, or reject, making access provisioning faster and more efficient.

Account Profiles in Oracle Access Governance

Feature	Description
Account Profiles in Oracle Access Governance	Oracle Access Governance simplifies permission management by letting you to define account profiles with supported or custom account attributes and default values. This avoids the need to repeatedly enter the account details required for provisioning in each Access Bundle. While defining account profiles, you may choose to provide default values or choose to ask the requester to provide values during the self-service request. For Policy-Based Access Control (PBAC), default values are used. You can associate an account profile to an access bundle to ensure consistent attribute application and easier updates.

Global Key Values in Oracle Access Governance

Feature	Description
Global Key Values	**Global Key Values** in Oracle Access Governance is a set of key-value pair `{label,value}` containing keys for identity or account attributes with values — such as project codes with project names, language codes with languages, and so on. This simplifies transformation or account management operations. You can import the defined key-value pairs using a CSV file and use these values across orchestrated systems. For example, you may import project codes with project names in a CSV file and use the values to derive value in inbound or outbound transformations and use it across your integrations. You can also use this to source value to custom account attributes.

Account Attributes for Account Management and Transformations in Oracle Access Governance

Feature	Description
Account Attributes for Account Management and Transformations in Oracle Access Governance	**Account Attributes** in Oracle Access Governance enables administrators to configure custom account attributes beyond the default account attributes supported for an orchestrated system, providing flexibility over account management operations. You can source values of these additional account attributes from the Managed System, from Global key values reference file, or define it when creating an access bundle. You can use and configure these attributes for inbound or outbound transformations, or for account provisioning operations, such as account creation. You can also use these account attributes to define the account profile required for provisioning. For example, you can dynamically construct nested attributes containing array of values, such as address, and use it during provisioning access to a user.

Integrate Oracle Access Governance with

Feature	Description
Integrate with	: Oracle Access Governance now supports identity orchestration for provisioning and reconciliation of accounts in as a Managed System. is a comprehensive solution managing privileged access across various IT environments, ensuring that only authorized users can access critical systems. With this integration, you can centrally manage and monitor privileged accesses, streamline access reviews, enforces security policies across all systems, ensuring compliance with internal and regulatory requirements.

New Orchestrated System: and

Feature	Description
Integrate with	: Oracle Access Governance now supports on-boarding of identity (user) data, identity orchestration for provisioning and reconciliation of accounts in as an Authoritative Source and as a Managed System.
Integrate with	: Oracle Access Governance now supports on-boarding of identity (user) data, identity orchestration for provisioning and reconciliation of accounts in as an Authoritative Source and as a Managed System.

ePrescribe, PPR and Taxonomy Attribute Support in Oracle Health EHR (formerly Cerner Millenium) Orchestrated Systems

Feature	Description
Extending Support of Default Attributes for Oracle Health EHR (formerly Cerner Millenium) Orchestrated System	Oracle Health EHR (formerly Cerner Millenium) Orchestrated System now supports additional default attributes for provisioning and transformation to enable seamless data integration. Oracle Health EHR (formerly Cerner Millenium) Orchestrated system now supports account attributes from the following functionalities : • ePrescribe • Taxonomy • Patient-Provider Relationship (PPR) Clinical Decision Support

New Service Helpdesk Administrator role in Oracle Access Governance

Feature	Description
New Service Helpdesk Administrator role for Advanced Administration in Oracle Access Governance	A new application role **AG_ServiceDesk_Admin** is introduced in Oracle Access Governance. This role empowers administrators to modify accounts and perform other advanced administrative functions directly in the Oracle Access Governance Console. Users with role can enable, disable, or delete accounts and can revoke permissions managed by Oracle Access Governance. Furthermore, users with this role can retry provisioning for failed or pending accesses.

Account Lifecycle Management Operations in Oracle Access Governance

Feature	Description
Account Lifecycle Management in Oracle Access Governance	Oracle Access Governance users with **AG_ServiceDesk_Admin** role can now directly perform the following operations from the **Manage Identities** page. • Suspend all the accounts and accesses for an identity at once, that have not been assigned directly in the Managed System, using the **Terminate** operation. Based on the account settings configured for your orchestrated system, account may be deleted or disabled. Once terminated, no accounts and associated accesses for that identity can be managed from Oracle Access Governance. You may again re-provision the accounts and the accesses, granted through policies (**Grant Type** *Policy*), using the **Activate** button, if required. • Revoke one or more permissions assigned directly from the Managed System or provisioned through request. • Retry provisioning of permissions with the **Failed** or **Pending** statuses. It is applicable for the permissions provisioned within Oracle Access Governance (**Grant type** as *Request* or *Policy*) • Disable or Delete one or multiple accounts managed by Oracle Access Governance. Once disabled all the associated accesses are removed. The accounts are still managed by Oracle Access Governance . You may re-provision the accounts and the accesses using the **Enable account** operation, if required. For deleted accounts, all the associated accesses are removed and you can no longer manage the accounts from Oracle Access Governance.

December 2024 Update

Publish Initial Data Event of Day-0 to Object Storage OCI Buckets

Feature	Description
Event Data Publisher	Event Data Publisher is now enhanced to export the initial data event (Day 0) to OCI Buckets irrespective of file size. The publishing status for the Day 0 export is sent as stream messages to OCI Streams. The incremental ongoing data events (Day N) will still be published based on file size, either to OCI Buckets or OCI Streams.

Orchestrated Systems

Feature	Description
New Orchestrated System: Integrate with	: Oracle Access Governance now supports identity orchestration for provisioning and reconciliation of accounts in as a Managed System. is a comprehensive cloud-based procurement and spend management service that helps businesses streamline and optimize their procurement processes, from sourcing to payment. With this integration, you can create, update, enable, and disable identity accounts. You can manage group assignment for identities using Access Bundles from Oracle Access Governance. For more details, see Integrate with SAP Ariba.
New Orchestrated System: Integrate with	: Oracle Access Governance now supports identity orchestration for provisioning and reconciliation of accounts in as a Managed System. is an Enterprise resource planning (ERP) platform built to help businesses run real-time analytics and simplify complex business processes such as order-to-cash, procure-to-pay, plan-to-product, and request-to-service. With this integration, you can enable and disable identity accounts. You can manage role assignment for identities using Access Bundles from Oracle Access Governance. You can also update account by locking or unlocking identity accounts.

November 2024 Update

Preventive Segregation of Duties (SOD) Analysis using

Feature	Description
Segregation of Duties (SOD) Analysis for Oracle Fusion Cloud Applications	Oracle Access Governance now supports preventive SOD checks through . With this update, Oracle Access Governance raises potential conflicts as part of access request approval task. Currently, the SOD violations check is scoped for Oracle Access Governance Access Bundles. For more details, see Manage Approvals.

Access Controls: Manage Assignment of OCI Cloud Services Application Roles

Feature	Description
Assign users to OCI cloud service application roles from Oracle Access Governance	You can now assign OCI cloud services application roles to identities with Oracle Access Governance. For this, package one or more OCI cloud services application roles in an access bundle, and assign it to users through a policy or an access request. You may further run identity access reviews for these assignments, if these are granted through user request. For more details, see Create an Access Bundle.

Identity Access Reviews for OCI Permissions managed by Oracle Access Governance

Feature	Description
Run Identity Access Reviews for OCI Permissions managed by Oracle Access Governance	For assignments managed by Oracle Access Governance, you can certify identities assigned to OCI IAM groups and OCI cloud services application roles as part of OCI Access Bundles reviews (Grant Type as `REQUEST`). For more details, see Review Accesses to Cloud Services Managed by Oracle Cloud Infrastructure (OCI).

Event-Driven Incremental (Day-N) Data Load for Oracle Identity Governance (OIG) Orchestrated System

Feature	Description
Event-Driven Incremental Data Load for OIG	Oracle Access Governance now supports both event-driven and periodic snapshot-based incremental data load for OIG Orchestrated System. For event-driven data load, you can enable a new option, **Do you want to enable OIG database incremental data load?**, to load data automatically based on occurrence of specific system events or changelog, ensuring real-time updates. To enable this feature, the database user must be granted required privileges.

Data Load Settings for Orchestrated Systems

Feature	Description
Data Load Settings for Orchestrated Systems	You can now set how often data should be loaded and updated between Oracle Access Governance and orchestrated systems. You can configure the timing and frequency for all orchestrated system except for Flat File and Oracle Cloud Infrastructure (OCI IAM). For more details, see Configure Data Load Schedule Settings for Orchestrated Systems.

September/October 2024 Update

Event Data Publisher in Oracle Access Governance

Feature	Description
Event Data Publisher in Oracle Access Governance	With Oracle Access Governance, you have the flexibility to export and continually publish data events to your cloud tenancy. You can export one-time and sequentially and continually publish ongoing data events to OCI Buckets or OCI Streams depending on the file size. See Event Data Publisher in Oracle Access Governance.

Orchestrated Systems

Feature	Description
Database Application Tables (Oracle) and User Management	1. Oracle Access Governance now supports the following: • Oracle Autonomous Database • Oracle Database 23ai, 19c, 18c or 12c as a single database, pluggable database (PDB), or Oracle RAC implementation 2. Oracle Access Governance now supports wallet-based authentication, in addition to basic authentication. To enable this, download the autonomous database wallet to your agent host, and then configure the **Easy Connect URL for Database** field in the orchestrated system. For more details, see Configure Wallet for Autonomous Database Integration .
New Orchestrated System: Integrate with Oracle Health EHR (formerly Cerner Millenium)	**Oracle Health EHR (formerly Cerner Millenium)**: You can enable identity orchestration for provisioning of accounts in Oracle Health EHR (formerly Cerner Millenium) as a Managed System. See Oracle Health EHR (formerly Cerner Millennium) Integration Reference.

Delegations

Feature	Description
Delegations	• Oracle Access Governance Administrator (`AG_Administrator`) can now manage delegations on behalf of Oracle Access Governance users. User Managers user can now update delegations for users they manage directly. • To manage delegation settings, you can access delegations from multiple paths within the Oracle Access Governance Console. See Manage Delegation Preferences.

Microsoft Entra ID Group Management

Feature	Description
Microsoft Entra ID Group Management	You can now manage group for Microsoft Entra ID. Oracle Access Governance supports provisioning of Security Group and Office Group using the Identity Collections functionality.

August 2024 Update

New Application Roles in Oracle Access Governance

Feature	Description
New Application Roles related to Orchestrated System	New Application Owner Roles introduced for Orchestrated System: • **`AG_AppOwner_Admin`**: Can create, manage, and configure all the integrations as part of Orchestrated systems. See Application Owner Administrator. • **`AG_AppOwner_Admin_Restricted`**: Can create new integrations with other systems by adding an Orchestrated system but manage and configure the integrations or resources that they own, as a resource owner. See Application Owner Restricted Administrator. See Predefined Application Roles Reference listing all application roles.
New Application Roles related to Access Controls	New Access Control Restricted Administrator Role introduced for Access Controls: • **`AG_AccessControl_Admin_Restricted`**: Can create all the resources included in the Access Control module, such as Roles, Identity Collections, Policies, Approval Workflows, Access Bundles, and Organizations. However, they can manage only the integrations or resources that they own, as a resource owner. See Access Control Restricted Administrator. See Predefined Application Roles Reference listing all application roles.

Run Ownership Reviews and Identity Access Reviews based on Direct Permissions

Feature	Description
Ownership Reviews	You can schedule campaigns to review ownership of resources that are created within Oracle Access Governance, either periodically or on an ad hoc basis. By performing this review, you can ensure accountability of resources lies only with the designated owners. See Resource Ownership.
Run Identity Access Reviews for directly assigned permissions	You can now quickly certify privileges for all Orchestrated systems based on the permissions ingested directly (`DIRECT`) from a Managed System without provisioning it first from Oracle Access Governance. See Identity Access Reviews based on Permissions Assigned Directly in Managed Systems.

Add Resource Owners

Feature	Description
Add Primary and Additional Owners for Orchestrated Systems, Access Control Resources, and Organizations	You can now add primary and additional owners for Oracle Access Governance resources. Any Oracle Access Governance active identity can be assigned as the resource owner. All the owners can read, update, or delete the resources that they own. See Add Primary and Additional Owners.

Orchestrated Systems

Feature	Description
New Orchestrated System: Integrate with Oracle Fusion Cloud Applications	**Oracle Fusion Cloud Applications**: You can enable identity orchestration, including on-boarding of identity (user) data, and provisioning of accounts for Oracle Human Capital (HCM) and Oracle Enterprise Resource Planning (ERP) accounts. This includes using Oracle Fusion Cloud Applications as an Authoritative source and as a Managed System for account provisioning. See Integrate with Fusion Cloud Applications.
New Orchestrated System: Integrate with Database Application Tables (Oracle)	**Database Application Tables (Oracle)**: You can enable identity orchestration, including on-boarding of identity (user) data, and provisioning of accounts for Database Application Tables (Oracle) both as an Authoritative source and as a Managed System. See Integrate with Database Application Tables (Oracle).
Configure Account Settings	You can now configure the account settings to support the Joiners, Movers, and Leavers process for your Orchestrated system. You can configure to send email to user or user manager when a new account is created. You can also choose to either disable or delete the account whenever an identity moves within or leaves your enterprise.

Access Controls

Feature	Description
Provisioning Users to OCI IAM groups from Oracle Access Governance	You can now provision users to OCI IAM groups from Oracle Access Governance. You can package multiple OCI IAM groups in an access bundle, and provision it to users through a policy or an access request.
Identity Lifecycle - Joiners, Movers, Leavers Process	New article describing automated provisioning for Joiners, Movers, and Leavers (JML) process in Oracle Access Governance. See Identity Lifecycle Management .

New Articles for My Access and Language Support

Feature	Description
Self-Service - My Access	New article on viewing your access details and managing your accounts in Oracle Access Governance. See View Access Details and Manage Account.
Language Support in Oracle Access Governance	New article that lists various languages supported by Oracle Access Governance Console and steps to update your browser's locale settings. See Supported Languages in Oracle Access Governance.

July 2024 Update

Access Reviews

Feature	Description
Access Reviews Fallback Mechanism	New fallback process is introduced to assign a valid reviewer or a campaign owner whenever an invalid reviewer or an invalid campaign owner is detected. This will prevent sudden termination of a campaign.

Feature	Description
New or Updated Access Reviews Articles	New or updated articles for Access Reviews: • Access Reviews in Oracle Access Governance - Certify Access Privileges with Campaigns and Event-Driven Micro Certifications • Working with Access Review Campaigns • Create Identity Access Review Campaigns • Create Policy Review Campaigns • Create Identity Collection Review Campaigns • Manage and Monitor Access Review Campaigns • Micro-Certifications: Event Driven Access Reviews • Configure and Manage Event-based Access Reviews • Understanding Reviewer's Actions for Effective Access Reviews • Perform Access Reviews - Evaluate and Certify Access Review Tasks

June 2024 Update

Orchestrated Systems

Feature	Description
Integrate with Orchestrated Systems	: You can now perform identity reconciliation, user management, and role assignment with integration.

May 2024 Update

Who Has Access to What

Feature	Description
Who Has Access to What	**Enterprise-wide Browser** As an *Enterprise-wide Access Administrator* or *Administrator*, get a comprehensive and centralized view of access information across your enterprise. Enterprise-wide Browser allows you to: • Browse through access information using various perspective views, such as identities, identity collections, roles, permissions, policies, resources, and organizations. • Use search capabilities and advanced filters to optimize your search query and locate specific access information. • Customize the default access profile layout by hiding or showing columns or reordering columns for a better user experience. • Run user-created identity and access control reviews and view the access review report. • Download the CSV file for the first 500 records available in the access profile view or download the PDF screenshot of the access detail.
Who Has Access to What	**My Access** As an Oracle Access Governance user, you can view access profile details in the self service section. Go to **My Stuff** → **My Access** to view your access details. The account details visible on the **My Accounts** page is now available on the **My Access** → **Accounts** page.

Notifications

Feature	Description
Notifications	The following enhancements have been added to notifications: • Notification delivery service: The ability to define an alternative to the default Oracle Access Governance notification email delivery service has been added. You can now configure an OCI email delivery service as an alternative. Refer to Configure an OCI Email Delivery Service for Notifications for details. • Recipient for Orchestrated System related notifications: You can now define an identity or email to act as the recipient for notifications relating to Orchestrated System operations. See Configure Identity/Email For Orchestrated System Related Notifications for details.

OCI Data Handling

Feature	Description
OCI Data Handling	The **Identity Attributes** functionality has been enhanced to provide the ability to define which OCI domain Oracle Access Governance should use as the source of truth when ingesting identity data from a multi-domain OCI instance.

Integrations

Feature	Description
System Integration	: You can now perform identity reconciliation, user management, and role assignment with integration.
Integrations	Updated Data Transformation topic within the Integration documentation.

March/April 2024 Update

Orchestrated Systems

Feature	Description
Integrate with Orchestrated Systems	• EntraID: Configuration has been updated to allow for certificate-based authentication, in addition to existing client secret authentication. • Oracle Identity Governance: Configuration of the OIG Orchestrated System now includes data filters to limit the data transported and ingested from Oracle Identity Governance.
Integrate with Orchestrated Systems	Integration documentation for the following managed systems has been updated: • Oracle Identity Governance Agent: Additional prerequisites added. Troubleshooting section added.

Integration

Feature	Description
New/Updated Integration Articles	New/updated articles for integration: • Identity Orchestration Overview • Identity Orchestration Components • Identity Orchestration Process Flow • Manage Oracle Access Governance Agent for Indirect Integrations • Manage Integrations with Orchestrated System • Configure Settings for an Orchestrated System • Supported Integrations with Oracle Access Governance • Data Rules to Customize and Transform Identity and Account Attributes

Feature	Description
Integration Landing Page	Integration landing page has been updated: • The Integration landing page has been redesigned to include new integration articles, and to provide a drop down list for each specific integrations which, when selected, will display all relevant content relating to the chosen integration system.

February 2024 Update

Orchestrated Systems

Feature	Description
Integrate with Orchestrated Systems	You can now integrate Oracle Access Governance with: • **Generic REST**: You can perform user management and teams group assignment tasks via Oracle Access Governance. • **Oracle Siebel**: You can perform user management and role grant management operations via Oracle Access Governance.

Unmatched Accounts

Feature	Description
Delete Unmatched Accounts	You now have the option to delete accounts which are unmatched from a Managed System. This is in addtion to the current functionality allowing you to match an unmatched account to an identity.

Configurable Notifications

Feature	Description
Configurable Notifications	You can now customize and configure notifications. Notifications are sent for different types of event. You can now customize notifications in the following ways: • Set the default logo • Set the default language in which notifications are sent • Enable notification type • Disable notification type • Set Subject for the notification email • Set content for the notification email body

Data Transformation and Matching Rules

Feature	Description
Identity Attributes	For custom identity attributes, you now have the option to add a rule on how the attribute is populated. You can either use the value directly, or you can create a rule around the active value.

January 2024 Update

Orchestrated Systems

Feature	Description
Integrate with Orchestrated Systems	You can now integrate Oracle Access Governance with: • Microsoft Teams: You can perform user management and teams group assignment tasks via Oracle Access Governance. • Oracle Primavera: You can perform user management and role grant management operations via Oracle Access Governance.

Data Transformation and Matching Rules

Feature	Description
Outbound and Inbound Data Transformation	You can transform the data coming into Oracle Access Governance or going out (provisioned) of Oracle Access Governance. You can apply transformation rules on the inbound and the outbound data, by writing methods in JavaScript, for objects, identity (user) object, account object, and custom user attributes.
Matching Rules	You can now use matching or correlation rules to avoid orphan or unmatched accounts during the data ingestion process. You can set up these rules to match the identity data imported from different authoritative sources, and/or match multiple accounts with an identity to avoid unmatched account.

December 2023 Update

Orchestrated Systems

Feature	Description
Integrate with Orchestrated Systems	You can now integrate Oracle Access Governance with Oracle Fusion Cloud Applications. With this integration, you can perform User management and Role grant management operations through Oracle Access Governance.

Outbound Data Transformations

Feature	Description
Outbound Data Transformation	Through Oracle Access Governance, you can now perform data transformation on the data provisioned into the Orchestrated system account.

Identity Collection can manage a new or existing Active Directory group on a Orchestrated System

Feature	Description
Identity Collection can manage a new or existing Active Directory group on a Orchestrated System	While creating an identity collection in Oracle Access Governance, you can now opt to manage a group on a Orchestrated system. The selected (new or existing) group in this Orchestrated system will be managed by this identity collection.

Reassignment of Identity and Access Reviews

Feature	Description
Reassign Identity and Access Reviews	Oracle Access Governance gives you the provision to reassign identity reviews and/or access review items to other users. In reassignment, the review tasks will be moved from the original reviewer and gets assigned to the new reviewer.

Access Governance Organization

Feature	Description
Oracle Access Governance Organizations	Oracle Access Governance administrators can now structure identities and form relationships between identities by creating and managing Organizations in the Oracle Access Governance Console.

Approval Workflows in the Event-Based Access Reviews

Feature	Description
Approval Workflows in the Event-based Access Reviews	You can now configure approval workflows for all the three event types - change event, timeline event and multi-event access reviews.

Unmatched Accounts Access Reviews

Feature	Description
Unmatched Accounts Access Reviews	You can now review unmatched accounts via event-based access reviews. This allows application owners and custom users to match an unmatched account to an existing Oracle Access Governance identity, or remove the account from the Orchestrated system.

Enhancement in Reporting of Campaign Details

Feature	Description
Campaign Details' Report Enhancement	In the campaign details page, for approval workflow summary, you can see count of total and pending reviews. A new link, **View pending link**, has been added that provides reviewer details, such as reviewer's name, email addresses, and count of pending reviews with each of them.

November 2023 Update

Orchestrated Systems

Feature	Description
Orchestrated Systems	The following types of Orchestrated System have been added to Oracle Access Governance: • Eloqua • NetSuite • Microsoft SQL Server • Microsoft Entra ID (formerly Microsoft Azure Active Directory • Flat File
Unmatched Accounts	The ability to manage unmatched accounts has been added. You can search for unmatched accounts, and where appropriate, match them to an Oracle Access Governance identity.
OCI Group Membership Review	OCI IAM group memberships can now be reviewed as part of Identity Collection access reviews.
Timeline Event Based Micro-certification	Timeline based micro-certification to trigger user access reviews based on specific dates, such as anniversary dates, has been added to the Event-Based Access Review functionality.
Active Directory group management	AD groups can now be managed from Oracle Access Governance using the Identity Collections functionality.

September 2023 Update

Time-based Events

Feature	Description
Time-based Events	Time-based Events refer to an event which is raised for a particular date, for example, weekly, monthly, or when a user is granted access to an application on a given date, which is subject to an annual review. A review task is generated for the user on the date configured for the time-based event, to determine if the permission associated with the event is still appropriate

July 2023 Update

List Requests that needs Approvals

Feature	Description
Approvals	The **Approvals** page in the Oracle Access Governance console, lists access requests requiring your attention. All requests requiring approval will be displayed. These requests are listed as one access per row. If a request is made for multiple accesses, for example access to a database, a directory, and a cloud service, then this will be displayed as 3 rows requiring separate approvals in your approval list.

Viewing Access Requests

Feature	Description
View My Requests	The **My Access Requests** screen, in the Oracle Access Governance console displays a list of access request raised by the logged-in user for Self or for others. You can either view the details, cancel a request or can provide information on the requests.

Request Access for Yourself or for Other Users

Feature	Description
Request Access	As an Oracle Access Governance user you can request access to resources and roles. Requests can be made for yourself, or for others. This process creates an access request which is subject to an approval workflow.

Simplifying Process of Requesting Resource Permission

Feature	Description
Create and Manage Access Bundle	An Access Bundle is a collection of permissions that packages access to resources, application features, and functionality into a requestable unit. To access a particular resource, you do not have to request each permission associated with that resource individually, instead you request an access bundle containing all permissions associated with that resource. This simplifies the process of requesting resource permissions. Using Oracle Access Governance console, you can now create a new access bundle and manage it.

Maintain Policies within your Oracle Access Governance Service

Feature	Description
Manage Polices	Using policies you can now provide access to resources within your organization. These policies associate resources and permissions with identities by means of roles and access bundles. You can create and manage policies by using Oracle Access Governance Console.

Manage Roles

Feature	Description
Manage Roles	You can now create and manage roles using Oracle Access Governance console. These roles are a group of access bundles. The access bundles contained within a role can span multiple targets. For example, a Database Administrator role groups together the DBAdmin_Oracle, DBAdmin_DB2, and DBAdmin_MySQL access bundles. To use a role you must associate identities to it via a policy.

Create and Manage Approval Workflow

Feature	Description
Create and Manage Approval Workflow	In Oracle Access Governance, every permission, access request, or role that needs to be assigned to a user must be processed through an approval workflow. You as a resource administrator can design an approval workflow by specifying the required approval level and the number of approvers. Later, as a Permission Manager you can use these workflows to obtain approvals before assigning or revoking user privileges. You as a resource administrator can monitor and manage the approvals using the Oracle Access Governance Console.

Integrate Oracle Access Governance with Target Systems

Feature	Description
Integrate with Orchestrated Systems	You can now connect Oracle Access Governance with the following systems by entering connection details and credentials for the target system. • Active Directory • Oracle e-Business User Management (UM) • Oracle e-Business Employee Reconciliation (HRMS) • Database User Management (Oracle) • Oracle Unified Directory • Oracle Internet Directory • Database User Management (MySQL) • Database User Management (DB2)

May 2023 Update

New License Types for Oracle Access Governance

Feature	Description
New License Types	Oracle Access Governance rolls out new license types for its users: • Access Governance for Oracle Cloud Infrastructure • Access Governance for Oracle Workloads

Added Identity Activation Rules for License Management in Oracle Access Governance

Feature	Description
Manage Identities	You can now optimize Oracle Access Governance instance operating cost by managing which identities can use the Oracle Access Governance service. Identities excluded from the service will not have access to Oracle Access Governance functionality and will not be billed.

Identity Orchestration: Integrate Oracle Access Governance with Oracle Cloud Infrastructure Identity and Access Management (OCI IAM)

Feature	Description
Integrate with OCI IAM	You can now implement code-less integration of Oracle Access Governance directly with cloud services. This release supports Identity Orchestration set up between Oracle Access Governance and Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) system.

Policy Reviews

Feature	Description
Policy Reviews	You can review OCI IAM policies either one-time or periodically from Oracle Access Governance by creating Policy Review campaigns. In this campaign, access control of each cloud resource is evaluated up to the policy statement-level. The policy statements can either be accepted or revoked.

Who Has Access to What: Enterprise-wide Access and Individual's Access to Cloud Resources

Feature	Description
Who Has Access to What	The Who Has Access to What capability now includes: • Ability to see individual's access to cloud resources. On the **My Access** page, you can now select a specific option from the **Group by** drop-down to view access to cloud resources assigned to you. • 360-degree visibility into organization's cloud resources, identities who can access these resources, and assigned permissions. Here, you can view a comprehensive list of all resources across various systems or cloud tenancies Orchestrated with Oracle Access Governance.

New Capability that supports Custom Identity Attributes in Oracle Access Governance

Feature	Description
Custom Identity Attributes.	Oracle Access Governance now supports custom identity attributes in addition to core identity attributes for running various Oracle Access Governance operations.

Introduced Identity Collections functionality in Oracle Access Governance

Feature	Description
Identity Collection	You can now create and manage a collection of identities to perform Oracle Access Governance functions collectively on a group rather than performing for each individual identity. You can create an identity collection either by defining conditional rules, known as Membership Rules, and/or by directly selecting identity names.

Added Capability to Delegate your Access Review Tasks to an Identity or an Identity Collection

Feature	Description
Delegation	Oracle Access Governance now provides the capability to delegate your tasks by setting up your preferences. In this release, from the **My Preferences** screen, you can delegate your access review tasks to an identity or to an identity collection.

> **Note:**
>
> For this release, you must upgrade your current Oracle Access Governance agent to enable code-less integration with the systems. Refer Agent Example Usage to enable the auto-upgrade flag and upgrade your agent with the latest updates.

February 2023 Update

New Available Region in Central UAE: Abu Dhabi

Feature	Description
Available in UAE Central: Abu Dhabi	Oracle Access Governance rolls out its services and is now available in the **UAE Central Abu Dhabi** region.

New Enterprise-wide Access functionality in Who Has Access to What

Feature	Description
Who Has Access to What	The **Who Has Access to What** capability now includes 360-degree visibility into organization resources, resource types, identities who can access these resources, and assigned permissions. Here, you can view a comprehensive list of all resources across various Orchestrated with Oracle Access Governance.

Auto Upgrade Feature for Oracle Access Governance Agent in Orchestrated Systems

Feature	Description
New Auto Upgrade Flag for Oracle Access Governance Agent in **Orchestrated Systems**.	You can now automatically install updates for the Oracle Access Governance Agent by enabling the *autoupgrade* flag during the configuration process. Through this flag, a scheduled task is run every 24 hours that checks and/or installs any updates available for the Oracle Access Governance agent. This is a crucial step and you must set this to prevent any issues in communication from the agent to the Access Governance Service. Refer Agent Example Usage to enable the auto-upgrade flag.

October 2022 Update

What's New in the October 2022 Update

Event-based Access Reviews

Feature	Description
Event-based Access Reviews	You can now launch event-based access reviews from Oracle Access Governance that initiate whenever a change is detected in a user lifecycle state or a user attribute, such as onboarding of new users, department change, job-code change, location change, retirement or exit of users, or manager change. Once configured, these are automatically triggered when one or more predefined event types occur.

Access Review Scheduler

Feature	Description
Access Review Scheduler	You can now schedule and run the Access Review Campaigns periodically which can be Monthly, Quarterly, Half-Yearly, or Yearly.

June 2022 Update

On-Demand Access Reviews

Feature	Description
On-Demand Access Reviews	You can launch on-demand Access Review Campaigns to review user access assignments where individual access to a specific source is checked and either certified or remediated.

Who Has Access to What

Feature	Description
Who Has Access to What	You can use the **Who Has Access to What** functionality as a user or a user manager to see the number of applications, permissions, and roles assigned to you (self) or to your direct reports.

Identity Orchestration in Oracle Access Governance

Feature	Description
Identity Orchestration	Oracle Access Governance enables code-less integration with on-premises and cloud systems. You can now configure Identity Orchestration in Oracle Access Governance Console. This release supports Identity Orchestration set up between Oracle Access Governance and Oracle Identity Governance (OIG) system.

3
Administer

System Administration

You can create an Oracle Access Governance instance in the Oracle Cloud Infrastructure Console. The steps below show you how to create an instance and verify its operation.

> **Note:**
>
> Oracle Access Governance is available in all the regions of the commercial realm. Full details about the regions can be referred to at Regions and Availability Domains.

Prerequisites

A prerequisite for creating and setting up a service instance is to provide permissions for **agcs-instance** resources.
To create an Oracle Access Governance service instance, the Oracle Cloud Infrastructure Identity and Access Management administrator or domain administrator can create a group and allow that group permissions to:

- Read *objectstorage-namespace* resources in tenancy (root compartment) in a policy statement.
- Manage *agcs-instance* resources for a given compartment or tenancy (root compartment) in a policy statement

To update or delete an Oracle Access Governance service instance, the Oracle Cloud Infrastructure Identity and Access Management administrator or domain administrator can create a group and allow that group permissions to:

- Manage *agcs-instance* resources for a given compartment or tenancy (root compartment) in a policy statement.

Example Policies for Tenancies using Identity Domains

1. Tenancy Admin

```
Allow group <domain_name>/<group_name> to manage all-resources in
    tenancy
```

2. Compartment Admin
   a. Add the following policy statement in the root compartment of your tenancy. This will fetch the tenancy namespace to create a service instance.

```
Allow group <domain_name>/<group_name> to read objectstorage-namespace in
    tenancy
```

b. Add the following policy statement in the compartment where you want create the service instance

```
Allow group <domain_name>/<group_name> to manage all-resources in compartment
    <compartment_name>
```

3. With ‘manage agcs-instance’ in tenancy

a. Add the following policy statement in the root compartment of your tenancy. This will fetch the tenancy namespace to create a service instance.

```
Allow group <domain_name>/<group_name> to read objectstorage-namespace in
    tenancy
```

b. Add the following policy statement in the compartment where you want create the service instance

```
Allow group <domain_name>/<group_name> to manage agcs-instance in
    tenancy
```

4. With ‘manage agcs-instance’ in a compartment

a. Add the following policy statement in the root compartment of your tenancy. This will fetch the tenancy namespace to create a service instance.

```
Allow group <domain_name>/<group_name> to read objectstorage-namespace in
    tenancy
```

b. Add the following policy statement in the compartment where you want create the service instance

```
Allow group <domain_name>/<group_name> to manage agcs-instance in compartment
    <compartment_name>
```

Example Policies for Tenancies without Identity Domains

1. Tenancy Admin

```
Allow group <group_name> to manage all-resources in
    tenancy
```

2. Compartment Admin

a. Add the following policy statement in the root compartment of your tenancy. This will fetch the tenancy namespace to create a service instance.

```
Allow group <group_name> to read objectstorage-namespace in
    tenancy
```

b. Add the following policy statement in the compartment where you want create the service instance

```
Allow group <group_name> to manage all-resources in compartment
    <compartment_name>
```

3. With ‘manage agcs-instance’ in tenancy

a. Add the following policy statement in the root compartment of your tenancy. This will fetch the tenancy namespace to create a service instance.

```
Allow group <group_name> to read objectstorage-namespace in
    tenancy
```

b. Add the following policy statement in the compartment where you want create the service instance

```
Allow group <group_name> to manage agcs-instance in
    tenancy
```

4. With ‘manage agcs-instance’ in a compartment
   a. Add the following policy statement in the root compartment of your tenancy. This will fetch the tenancy namespace to create a service instance.

      ```
      Allow group <group_name> to read objectstorage-namespace in
          tenancy
      ```

   b. Add the following policy statement in the compartment where you want create the service instance

      ```
      Allow group <group_name> to manage agcs-instance in compartment
          <compartment_name>
      ```

Create Service Instance

Create an Oracle Access Governance instance in the Oracle Cloud Infrastructure console.

You can create an Oracle Access Governance service instance using the following steps:

1. Open your web browser and navigate to https://cloud.oracle.com.
2. Enter the name of your **Cloud Account Administrator** in the **Cloud Account Name** field and click **Next**.
3. On the Cloud Infrastructure sign-in page, enter your sign-in credentials under **Oracle Cloud Infrastructure Direct Sign-In**. Click **Sign In**.
4. When you have successfully logged in, select **Regions → [US East (Ashburn)|Brazil East (Sao Paulo)|Germany Central (Frankfurt)|Australia East (Sydney)]**, depending on your Home region location, from the top navigation menu.
5. Click the  icon in the top left corner to display the navigation menu.
6. Click **Identity and Security** in the navigation menu.
7. Select **Access Governance** from the list of products.
8. On the **Service Instances** page, click the **Create service instance** button.
9. Enter values for the service instance as detailed in the following table .

Parameter	Value	Description
Name		Name of the service instance.
Description		Description of the service instance.
Create in compartment	**Compartment Name** into which the service instance will be created.	Name of the OCI compartment into which the service instance will be created.

Parameter	Value	Description
License type		Select from the following license types: • **Access Governance Premium**: Governance of access privileges for Oracle and Non-Oracle Workloads running anywhere • **Access Governance for Oracle Workloads**: Governance of access privileges for Oracle Workloads running anywhere • **Access Governance for Oracle Cloud Infrastructure**: Governance of access privileges for OCI resources and services. Access Governance for OCI is the entry level license option, covering OCI in cloud environments. Access Governance for Oracle Workloads is a broader option, covering Oracle Workloads running anywhere, and includes OCI. Access Governance Premium is the widest option, including non-Oracle as well as Oracle workloads. When you select a license option, be aware that it may take approximately 10 minutes before the licence is enabled on your service instance.
Tagging		Tags allow you to organize and track resources within your tenancy. If you want to tag resources within the service instance, add them here. Add value as described in the following rows. If you want to add additional tags, select **Another Tag** to create more.
TAG NAMESPACE		Namespace to which the tag applies.
TAG KEY		Key for the tag.
TAG VALUE		Value of the tag.

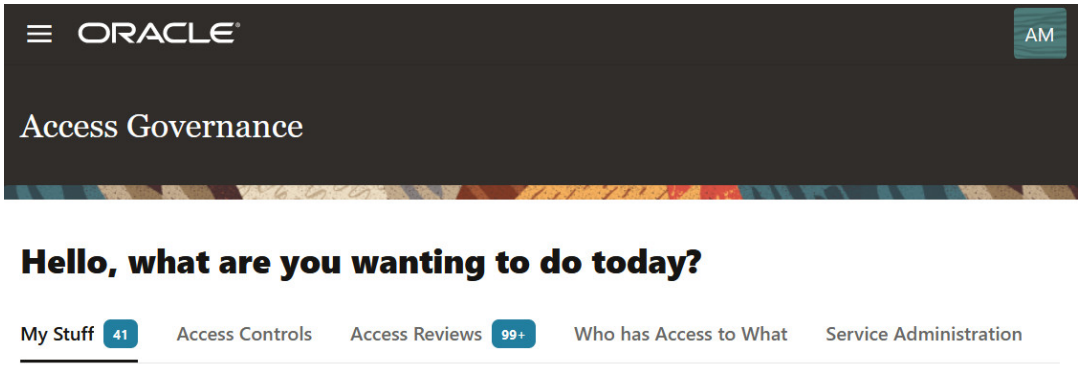
10. To create the service instance with the value you have input, select **Create service instance**. If you do not want to proceed with the service creation, select **Cancel**.

Verify Service Instance

You can verify an Oracle Access Governance service instance using the following steps:

1. Open your web browser and navigate to https://cloud.oracle.com.
2. Enter the name of your **Cloud Account Administrator** in the **Cloud Account Name** field and click **Next**.
3. On the Cloud Infrastructure sign-in page, enter your sign-in credentials under **Oracle Cloud Infrastructure Direct Sign-In**. Click **Sign In**.
4. Click the icon in the top left corner to display the navigation menu.
5. Click **Identity and Security** in the navigation menu
6. Select **Access Governance** from the list of products.
7. On the **Service Instances** page, select the newly created service instance.
8. Click **Service Home Page** to access the Oracle Access Governance Console in a browser.

   The Oracle Access Governance Home page should look similar to below. Depending on the application roles assigned to your user, you will see the following tabs:



   - **My Stuff**
   - **Access Controls**
   - **Access Reviews**
   - **Who Has Access to What**
   - **Service Administration**

   You can select which Oracle Access Governance task you want to perform by selecting the relevant tab, and clicking on the tile displayed for your task. Alternatively, you can select tasks from the navigation menu, .

You can manage an Oracle Access Governance instance in the Oracle Cloud Infrastructure Console. The steps below show you how to perform management tasks on your service instance using the Oracle Cloud Infrastructure Console.

Review Service Instance

As *Cloud Account Administrator*, you can review Oracle Access Governance instances in the Oracle Cloud Infrastructure Console.

To review the details of a service instance, use the tasks detailed in this section:

1. Open your web browser and navigate to https://cloud.oracle.com.
2. Enter the name of your **Cloud Account Administrator** in the **Cloud Account Name** field and click **Next**.
3. On the Cloud Infrastructure sign-in page, enter your sign-in credentials under **Oracle Cloud Infrastructure Direct Sign-In**. Click **Sign In**.
4. Click the  icon in the top left corner to display the navigation menu.
5. Click **Identity and Security** in the navigation menu
6. Select **Access Governance** from the list of products.
7. On the **Access Governance** page, select **Service Instances**.
8. Click the **Actions** menu for the service instance that you want to review.
9. Select **View details**.
10. Review the information for the service instance in the **Service Instance Details** page.

Launch Service Home Page

As a *Cloud Account Administrator*, you can launch the service home page for Oracle Access Governance from the Oracle Cloud Infrastructure Console.

To launch the service home page:

1. Open your web browser and navigate to https://cloud.oracle.com.
2. Enter the name of your **Cloud Account Administrator** in the **Cloud Account Name** field and click **Next**.
3. On the Cloud Infrastructure sign-in page, enter your sign-in credentials under **Oracle Cloud Infrastructure Direct Sign-In**. Click **Sign In**.
4. Click the  icon in the top left corner to display the navigation menu.
5. Click **Identity and Security** in the navigation menu
6. Select **Access Governance** from the list of products.
7. On the **Access Governance** page, select **Service Instances**.
8. Click the **Actions** menu for the service instance that you want to review.
9. Select **Service home page**.
10. Perform activities in the Oracle Access Governance Console.

Edit Service Instance

As a *Cloud Account Administrator*, you can edit an Oracle Access Governance service instance from the Oracle Cloud Infrastructure Console.

To edit a service instance:

1. Open your web browser and navigate to https://cloud.oracle.com.

2. Enter the name of your **Cloud Account Administrator** in the **Cloud Account Name** field and click **Next**.
3. On the Cloud Infrastructure sign-in page, enter your sign-in credentials under **Oracle Cloud Infrastructure Direct Sign-In**. Click **Sign In**.
4. Click the  icon in the top left corner to display the navigation menu.
5. Click **Identity and Security** in the navigation menu
6. Select **Access Governance** from the list of products.
7. On the **Access Governance** page, select **Service Instances**.
8. Click the **Actions** menu for the service instance that you want to review.
9. Select **Edit**.
10. The *Cloud Account Administrator* can update the name and description of the service selected, or upgrade the Licence Type as required.

    Licence type becomes more inclusive in the following order:

    a. **Access Governance for Oracle Cloud Infrastructure**
    b. **Access Governance for Oracle Workloads**
    c. **Access Governance Premium**

    You can only upgrade to a more inclusive licence type:

    - Access Governance for Oracle Cloud Infrastructure to Access Governance for Oracle Workloads is a valid upgrade.
    - Access Governance for Oracle Cloud Infrastructure to Access Governance Premium is a valid upgrade.
    - Access Governance for Oracle Workloads to Access Governance Premium is a valid upgrade

    You cannot downgrade to a less inclusive licence type:

    - Access Governance Premium to Access Governance for Oracle Workloads is not valid.
    - Access Governance Premium to Access Governance for Oracle Cloud Infrastructure is not valid.
    - Access Governance for Oracle Workloads to Access Governance for Oracle Cloud Infrastructure is not valid.

    When you select a license option, be aware that it may take approximately 10 minutes before the licence is enabled on your service instance.

Delete Service Instance

As a *Cloud Account Administrator*, you can delete an Oracle Access Governance service instance from the Oracle Cloud Infrastructure Console.

To delete a service instance:

1. Open your web browser and navigate to https://cloud.oracle.com.
2. Enter the name of your **Cloud Account Administrator** in the **Cloud Account Name** field and click **Next**.

3. On the Cloud Infrastructure sign-in page, enter your sign-in credentials under **Oracle Cloud Infrastructure Direct Sign-In**. Click **Sign In**.
4. Click the  icon in the top left corner to display the navigation menu.
5. Click **Identity and Security** in the navigation menu
6. Select **Access Governance** from the list of products.
7. On the **Access Governance** page, select **Service Instances**.
8. Click the **Actions** menu for the service instance that you want to review.
9. Select **Delete**.
10. The service instance is marked for deletion. The service URL to the Oracle Access Governance Console is inaccessible to all users.

> **Note:**
>
> The service instance is displayed in the OCI Console for a period of 60 days following deletion, with a status of *Deleted*. After 60 days the deleted service instance is removed from the OCI Console display.

Notification settings for Oracle Access Governance can be managed in the Oracle Cloud Infrastructure Console and Oracle Access Governance Console.

Oracle Access Governance will keep you informed of significant events occurring within your service instance via email notifications. Notifications are triggered by an event such as account creation, or are sent periodically, such as pending review tasks which are sent daily. Notification types include:

- Account operations: Actions such as account creation and account modification will trigger a notification.
- Approval operations: Actions such as approval assignment or approval escalation.
- Review tasks: Actions such as review task assignement or pending review tasks.
- Error alerts: Actions such as agent disconnected from orchestrated system or failed target operation notification for an orchestrated system.

Notifications are sent by email only, using the default Oracle Access Governance email server, or by your own OCI email delivery service. You can configure global settings such as notification email, logo, and language. Specific notification types can be enabled or disabled, and you can set notification email Subject, and the content for the notification email body.

Configure Notification Email Display Name Using OCI Console

You can set the **Display name** for the From field of the notification email in the Oracle Cloud Infrastructure Console.

The notification email for Oracle Access Governance is the sender email address that is used to send all notifications regarding campaigns to your users. The default mail address is `no-reply@access-governance.oci.oraclecloud.com`. Oracle Access Governance does not

currently support modification of the default mail address. To set the **Display name** for the notification email for Oracle Access Governance:

1. Log in to the Oracle Cloud Infrastructure Console as an administrator.
2. Click the  icon in the top left corner to display the navigation menu.
3. Click **Identity & Security** in the navigation menu.
4. Select **Access Governance** from the list of products.
5. Select **Settings** menu.
6. In the **Notification email** section, enter the following details:
   - **Display name**: Optionally, enter a display name for the sender's email address.

Configure an OCI Email Delivery Service for Notifications

By default, Oracle Access Governance uses its own email delivery service to send notifications. You can override the default server by configuring an alternative OCI email delivery service if required.

Specify your OCI email delivery service for notifications by carrying out the following steps:

1. From the Oracle Access Governance service home page click on the  icon, and select **Service Administration → Notifications**.
2. In the top right hand of the page, select **Manage notification service**.
3. From the **Manage notification service** panel, select **Yes** to configure your own service for notification delivery. Enter values for the following parameters:
   - What do you want to name this service?
   - What username should be used?
   - What password should be used?
   - What is the public endpoint?
   - Which port should be used?
   - What is the from email address?
4. Optionally, you can verify your configuration by sending a test email using the settings you have applied in the previous step. Input a test email address in the **Which address should we send the test email to?** field and click **Send test email** to test the connection. Check the email of the test user to confirm that the test email was sent and received.
5. If you are happy with the configuration, and your test address verifies the setup, select **Save** to save your settings.

Configure Global Settings for Notifications

You can use Oracle Access Governance Console to update global notification settings, including the logo used in the notification email, and the default language to use for notifications in your Oracle Access Governance service instance.

1. From the Oracle Access Governance service home page click on the  icon, and select **Service Administration → Notifications**.

2. Select one of the following global settings to update, from the **Logo and languages settings** drop-down.
   - To update the logo used in your notifications, select the **Change logo** link. In the **Manage logo** dialog, select a JPEG or PNG file as the source of your logo, and click **Save**.
   - To update the default language used in all notifications, select the **Change default language** link. In the **Manage default language** dialog, select the language you want to use as the default for notifications from the list of values, and click **Save**.

> **Note:**
>
> Your system default locale will be used to build the notification when no user specific locale has been detected. The user locale is detected by reading the locale setting from the browser session when the user logs into Oracle Access Governance.

Configure Notification Types

You can use Oracle Access Governance Console to update notification types, including enabling/disabling types, setting Subject for the notification email, and setting content for the email body.

1. From the Oracle Access Governance service home page click on the icon, and select **Service Administration** → **Notifications**.

2. Select the **Actions** menu, for one of the notification types in the **Notification types** drop-down.
3. Select one of the following from the **Actions** menu.
   - **Enable/Disable**: Select this to either enable or disable the notification type
   - **View details**: Select this to update Subject or email body for the notification type
4. If you selected **View details** in the previous step, then you are navigated to the settings page for the notification type selected, for example, *Account creation*. The language templates available for the notification type are listed. Select your language and click the edit icon. This will take you to the **Edit** page for your template.
5. Update the settings for the selected notification type.
   - Update the **Subject** with the value you want to display in the *Subject* field of the email for this notification type.
   - Update the content of the email by selecting **Download for customization**. Save the HTML file for the email body, edit with your changes, and then upload the modified file by clicking on **Upload customization**

> **Note:**
>
> the variables that can be applied for the Subject and email body are listed on the **Edit** page. Select **Show me the available variables** to show the variables you can use.

6. When you are happy with your changes, select **Save** to store your settings.

Configure Identities or Email for Sending Orchestrated System Related Notifications

If an issue occurs in an orchestrated system during dataload, you want to be notified in good time so that you can investigate and resolve the issue. You can configure identities or an external email, to route notifications regarding your orchestrated system to assist with this.

To send orchestrated system-related notifications to your preferred identities or an external email address, you can configure Oracle Access Governance as required:

1. From the Oracle Access Governance service home page click on the icon, and select **Service Administration → Orchestrated Systems**.
2. Select the orchestrated system you want to configure notifications for.
3. From the tiles in the **Configuration** section of the page, select **Manage** on the **Notification settings** tile.
4. In the **Which identities?** field, use the drop-down list to select identities in your Oracle Access Governance instance to send orchestrated system-related notifications to. You can have multiple identities as required.
5. In the **Email** field, add an email for any person external to your Oracle Access Governance instance (who does not have an identity in your system) who you would like to receive notifications. You can only add one external email address for orchestrated system-related notifications.

Service Administration

Administrators can manage two types of identity population within the Oracle Access Governance service. The **Manage Identities** feature allows administrators to activate/inactivate identities within the service, and flag identities as either *Workforce* or *Consumer* users.

Active/Inactive Identities

- **Active identities**: Identities flagged as active within the Oracle Access Governance service, which enables the following features:
  - Access to the Oracle Access Governance console, allowing identities to utilize features including My Access, My Access Reviews, My Preferences and so on.
  - Allows the identity's access to be governed in Oracle Access Governance.
  - Allows identities to be included in access review campaigns.
  - Active identities are considered for billing purposes.

- **Inactive identities**: Identities flagged as inactive within the Oracle Access Governance service.
  - Inactive identities have no access to the Oracle Access Governance console.
  - Inactive identities access governance is not governed in Oracle Access Governance.
  - Inactive identities are not included in access review campaigns.
  - Inactive identities are not considered for billing.

> **Note:**
>
> The default status of identities present in Oracle Access Governance is NULL. In order for identities to use the service functionality, and be considered for billing, you must activate all users for which this is required, using the steps detailed in this article.

Identities imported from Oracle Identity Governance have a status of **Disabled** or **Enabled**. This is different from the Oracle Access Governance status **Active/Inactive**. You should consider the following conditions when dealing with identities imported from Oracle Identity Governance:

- A Disabled identity can be marked as an Active identity in Access Governance to review its access privileges.
- An Oracle Access Governance Administrator may set rules, based on the attributes of disabled identites, to mark those disabled identities as Active in Oracle Access Governance.
- Oracle Access Governance will include only those Disabled identities for billing that are marked as Active.

Consumer/Workforce Users

A user can be either a Workforce user or a Consumer. The main difference is that a Consumer user has no access to the Oracle Access Governance service. By default, users are Workforce users. The specific differences between the two types are given in the table below:

Table 3-1 Workforce and Consumer Users

Capabilities	Workforce User	Consumer User
Access the Oracle Access Governance service: by console or programmatically.	YES	NO
Perform configurations and integrations, such as orchestrated systems, identity marking, identity attributes.	YES	NO
Manage access control objects (Role, Access Bundle, Identity Collection, Policy).	YES	NO
Manage access review campaigns (event-based, periodic, one-time).	YES	NO

Table 3-1 (Cont.) Workforce and Consumer Users

Capabilities	Workforce User	Consumer User
Generate reports for access reviews and approvals.	YES	NO
View access privileges assigned to self or others.	YES	NO
Raise access request for self and/or others.	YES	NO
Perform access approval tasks.	YES	NO
Access privileges are managed by others.	YES	YES
Assigned access privileges are assigned by others.	YES	YES

Navigate to Manage Identities

Here's how you can access the Manage Identities page:

1. Log in to the Oracle Access Governance Console as a user with the *Administrator* application role.
2. Click  in the top left corner to display the navigation menu.
3. Select **Service Administration** → **Manage Identities** to begin defining your identity rules.

The Manage Identities page is displayed, where you have to define which identities you want to activate. Oracle Access Governance identities are displayed in this page with each identity showing attributes such as First Name, Last Name, Employee User Name, Email, and others. You can modify the attributes displayed for each identity by selecting the **Edit list settings**  icon. In the **List settings** pop-up, you can choose to *Show* or *Hide* attributes. An example would be that you want to flag identities which have delegations defined. To implement this you would select to *Show* the *Delegation* attribute.

You can use the Search field to locate the required identity using a string search. Alternatively you can select one of the available filters, for example, if you select the *Delegation* ***Yes*** filter would restrict identities displayed to those for which delegations are defined.

Select Identities for Activation

In the **Manage Identities** page, an Administrator defines the identities that you want to include in the Oracle Access Governance service.

You can identify identities to include in your service by selecting criteria based on conditional statements. Either at least one (**Any**) or all (**All**) the set conditions must be satisfied. The list of available attributes is determined by the ingested data from the Managed System, and may include custom attributes.

You can select identities based on **Membership rule** and/or **Named identities**. Identities satisfying the set criteria for the Membership rule will automatically be included in your service. Using Named identities, you can directly add specific identities based on their full name.

You can also exclude specific members from your service by selecting **Manage exclusions** and entering the identities you want to exclude.

1. Select **Any** if any one of the set conditions should be satisfied, or select **All** if all the set conditions must be satisfied for that identity.
2. Select the attribute name from the list.

> **Note:**
>
> Based on the Managed System, you can select both core and/or custom attributes. To enable custom attributes, see Manage Identity Attributes

3. Select the conditional operator. Based on the data type of the attribute selected, the usage of these operators will vary.
4. Type the attribute value.
5. Continue to add the conditional statements or rules for more attributes.
   By default all the identities matching the criteria will be included. Click the **Manage Exclusions** button next to **Excluding # identity from the attribute conditions** and then select the identities that you want to exclude from your service.
6. Once you have defined your rules, select **Preview summary based on the rule above** to go to the *Preview Summary* popup. This will display the following information, for the top 10 in each category:
   - Total number of matches based on the rules you have entered.
   - Total number of identities in the service.
   - Breakdown of the distribution of included identities based on:
     - Organization
     - Job code
     - Location
     - Employee type
7. If you are satisfied with your preview, click **Save**.

> **Note:**
>
> Existing customers with identities loaded from Oracle Identity Governance should be aware that they must activate identities required, else they will not be able to see loaded identities in the system as all identities are excluded by default. Customers in this situation can either activate users, as described above, or set the following rule which will activate all identities they previously loaded from Oracle Identity Governance.
>
> ```
> status equals Active
> ```

Select Consumer Users

In the **Manage Identities** page, an Administrator defines the identities that you want to be flagged as consumer users in the Oracle Access Governance service.

You can identify identities to include as consumers in your service by selecting criteria based on conditional statements. Either at least one (**Any**) or all (**All**) the set conditions must be satisfied. The list of available attributes is determined by the ingested data from the Managed System, and may include custom attributes.

You can select identities based on **Membership rule** and/or **Named identities**. Identities satisfying the set criteria for the Membership rule will automatically be included as consumers in your service. Using Named identities, you can directly add specific identities based on their full name.

You can also exclude specific members from your service by selecting **Manage exclusions** and entering the identities you want to exclude.

1. Select **Any** if any one of the set conditions should be satisfied, or select **All** if all the set conditions must be satisfied for that identity.
2. Select the attribute name from the list.

> **Note:**
>
> Based on the Managed System, you can select both core and/or custom attributes. To enable custom attributes, see Manage Identity Attributes

3. Select the conditional operator. Based on the data type of the attribute selected, the usage of these operators will vary.
4. Type the attribute value.
5. Continue to add the conditional statements or rules for more attributes.
   By default all the identities matching the criteria will be included. Click the **Manage Exclusions** button next to **Excluding # identity from the attribute conditions** and then select the identities that you want to exclude from your service.
6. Once you have defined your rules, select **Preview summary based on the rule above** to go to the *Preview Summary* popup. This will display the following information, for the top 10 in each category:
   - Total number of matches based on the rules you have entered.
7. If you are satisfied with your preview, click **Save**.

Create and Manage Organizations

You can now structure identities and form relationships between identities by creating and managing Organization with the Oracle Access Governance Console.

You can use Organizations to perform various operations within the Oracle Access Governance Console. For example, you can use it as an attribute (*Organization*) to create an Identity Collection, which can then be used for identity reviews, assigning access privileges, or for provisioning operations.

> **Note:**
>
> This Organization concept is native to Oracle Access Governance and is different than the source organization, which is loaded from an orchestrated system. It will be available in the core attribute list as **agOrganization** *(where the orchestrated system is Internal)* with the **Manage Identities** flag set to true. See View and Configure Custom Identity Attributes. If this flag is set to true, you can use this Organization to create/manage an Identity collection within Oracle Access Governance.

In the Oracle Access Governance Console, click the  icon, and select **Service Administration**, and then **Manage Identities**, and then **Organizations**. You will see the **Organizations** page where you can view and manage existing organization, or create new ones.

Create Organization

To create a new organization, click the **Create an organization** button. The **Add Details** task is displayed. In the **Add Details** task, you can enter specifics about your organization. Here, you can give a meaningful name and add its supporting description.

1. Enter a name for your organization in the **What do you want to call this organization?** field.
2. Add a description for your organization in the **How would you describe this organization?** field.
3. Select one or more identities from the **Who else can manage this organization** list. The owner along with the listed identities can manage this organization.
4. Add one or more tags to identify or search your organization.
5. Once you have set your preferences, select **Next** to go to the *Select Identities* step.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Select Identities

In the **Select Identities** task, add identities that you want be part of your organization. You can select identities based on **Membership rule** and/or **Named identities**. For Membership rule, the identities satisfying the set criteria will automatically be included in organization. In Named identities, you can directly add identities based on their full name. All the available active identities (configured from the Licence Management page) will be displayed.

You can also exclude specific members from your organization by selecting **Manage exclusions** and entering the identities you want to exclude.

1. Select **Any** if any one of the set conditions should be satisfied, or select **All** if all the set conditions must be satisfied for that identity.
2. Select the attribute name from the list.

> **Note:**
>
> Based on the orchestrated system, you can select both core and/or custom attributes. To enable custom attributes, see View and Configure Custom Identity Attributes

3. Select the conditional operator. Based on the data type of the attribute selected, the usage of these operators will vary.
4. Type the attribute value.
5. Continue to add the conditional statements or rules for more attributes.
   By default all the identities matching the criteria will be included. Click the **Manage Exclusions** button next to **Excluding # identity from the attribute conditions** and then select the identities that you want to exclude from an organization.
6. Once you have set your preferences, select **Next** to go to the **Review and submit** step.
7. You can preview graphical summary of how many identities are included in your organization by clicking the **Preview the organization** link. This link is available on the right-side, towards the bottom of the **Who is included** panel.
8. If you are satisfied with your organization preview, click **Create**.

Manage Organization

Oracle Access Governance *Administrators* can view and manage organizations from the Oracle Access Governance Console. You can view existing organization and manage the ones that you created, or are authorized to manage, using the Oracle Access Governance Console.

Use the ••• **Actions** menu icon to edit, delete or view details of the organization.

> **Note:**
>
> Only organization owners and/or authorized users (selected while creating/modifying an identity collection) can edit or delete the organization.

You can perform the following:

- **Search and Filter available organizations**: You can use the **Search** field to locate the required organization by its name. You can narrow down the results by applying the available filters.

- **Edit an organization**: The **Edit an organization** page provides the same guided tasks as you see while creating a new identity collection. Owner of the organization and/or authorized users can modify its description, identity type, or added identities. After updating the details, on the **Review and submit** step, select **Update** to update the organization.
- **View organization details**: You can see **Organization** page displaying complete organization details, such as Organization owner, created and last modified dates, current members, as well as how the current members were included (through named identities or membership rule).
- **Delete an organization**: You can delete the organization if you are the owner of the organization or you have been given the rights by the owner. If an identity collection is based on the deleted organization value, then those identities would no longer be members of that identity collection.

Manage Account Lifecycle with Oracle Access Governance Service Desk Administrator Support

As a user with `AG_ServiceDesk_Admin` role, you can directly initiate account management operations with no approval process. You may enable, disable, delete accounts, or terminate all accounts and associated accesses for an identity. You can also retry provisioning for failed or pending statuses, and revoke permissions assigned directly or through requests from the **Manage Identities** → **Identities** page.

Additionally, `AG_ServiceDesk_Admin` can manage delegations or change password. For more details, see Manage Delegation Preferences and View Access Details and Manage Account.

Terminate all Accounts and Accesses for an Identity

You can terminate accounts and associated accesses for an identity immediately without an approval process. The identity would still remain **Active** in Oracle Access Governance.

Accounts and accesses with **Grant Type** `DIRECT` cannot be terminated. You may view the termination status by selecting the **Terminated in Access Governance** column from the **Manage Identities** → **Identities** page.

Required roles: `AG_ServiceDesk_Admin`
For more details, see Application Roles and Responsibilities Reference.

Terminate Accounts and Accesses for an Identity

1. Log in to Oracle Access Governance Console.
2. Click the navigation icon in the top left corner to display the navigation menu.
3. Select **Service Administration**, and then **Manage Identities**. The **Identities** page opens by default.
4. From the Identities list, select the **Actions** icon and select **Terminate**.

   A confirmation dialog box is displayed confirming that all accounts and accesses for an identity will be deleted or disabled based on account settings.
5. Select the acknowledgment check box to terminate an identity access.

6. Select **Terminate** in the dialog box. A confirmation message stating termination request has been initiated will be displayed.

   Once terminated, the status changes to **Yes** in the **Terminated in Access Governance** column along with a flag `Terminated` in the Identity details page. You may have to edit the list settings to view the column. You cannot manage these accounts from Oracle Access Governance.

Activate Accounts and Accesses for an Identity

You can re-provision terminated accounts and accesses using the Activate operation, ensuring seamless account management in Oracle Access Governance.

Terminated accounts with **Grant Type** `Policy` can be re-provisioned into Oracle Access Governance.

For more details, see Application Roles and Responsibilities Reference.

1. From the Identities list, select the **Actions** icon and select **Activate**.

   A confirmation dialog box is displayed confirming accesses for an identity will be enabled.
2. Select the acknowledgment check box to activate an identity access.
3. Select **Activate**. A confirmation message stating activation request has been initiated will be displayed.
4. Once activated, the status changes back to **No** in the **Terminated in Access Governance** column. You can now manage these accounts for an identity from Oracle Access Governance.

Revoke Permissions for an Account Managed by Oracle Access Governance

Permissions assigned directly, with grant type `Direct`, or access bundles granted through a self-service request, with grant type `Request`, can directly be revoked for an identity from the **Manage Identities** → **Identities** page.

You cannot revoke permissions for Oracle Cloud Infrastructure (OCI) and Oracle Identity Governance (OIG).

1. Log in to Oracle Access Governance Console.
2. Click the navigation icon in the top left corner to display the navigation menu.
3. Select **Service Administration**, and then **Manage Identities**. The **Identities** page opens by default.
4. From the Identities list, select the **Actions** icon and select **View details**.

   The Identity details page is displayed with the **Permissions** tab selected by default.
5. Select the **Actions** icon corresponding to the permission that you want to revoke.

6. Select **Revoke permission**. A confirmation dialog box is displayed confirming that permission accesses will be revoked.
7. Select **Revoke**. A confirmation message stating revocation request has been initiated will be displayed.

During the revocation process, a transitional state indicator with an icon and tool tip are displayed. Once the process is complete, the final status is displayed in the **Status** column. For failed or pending statuses, you may again retry provisioning for the permissions provisioned within Oracle Access Governance.

Retry Provisioning for Failed or Pending Accesses

You can retry provisioning for accesses with the **Failed** or **Pending** statuses. You can perform this operation for access bundles granted through request or granted via policy, that is, **Grant Type** *Request* or **Grant Type** *Policy*.

1. Log in to Oracle Access Governance Console.
2. Click the navigation icon in the top left corner to display the navigation menu.
3. Select **Service Administration**, and then **Manage Identities**. The **Identities** page opens by default.
4. From the Identities list, select the **Actions** icon and select **View details**.

   The Identity details page is displayed with the **Permissions** tab selected by default.
5. Select the **Actions** icon corresponding to the permission for which you want to retry provisioning.
6. Select **Retry provisioning**. A confirmation dialog box is displayed confirming that provisioning for the accesses will be retried.
7. Select **Retry**. A confirmation message will be displayed.

During the retry provisioning process, a transitional state indicator with an icon and tool tip are displayed. Once the process is complete, the final status is displayed in the **Status** column.

Disable and Enable an Account Managed by Oracle Access Governance

You can directly disable one or more accounts that are managed by Oracle Access Governance. You can perform this operation for the orchestrated systems that support this operation.

Disable an Account Managed by Oracle Access Governance

1. Log in to Oracle Access Governance Console.
2. Click the navigation icon in the top left corner to display the navigation menu.
3. Select **Service Administration**, and then **Manage Identities**. The **Identities** page opens by default.

4. From the Identities list, select the **Actions** icon and select **View details**.

   The Identity details page is displayed with the **Permissions** tab selected by default.

5. Select the **Accounts** tab.

6. Select the **Actions** icon corresponding to the account that you want to disable.

7. Select **Disable account**. A confirmation dialog box is displayed confirming that account will be disabled and access will be removed.

8. Select **Disable**. A success message is displayed.

   During the process, a transitional state indicator with an icon and tool tip are displayed. Once the process is complete, the final status changes to **Disabled** in the **Status** column.

Enable an Account to be Managed by Oracle Access Governance

9. To enable a disabled account and restore the accesses, select the **Actions** icon corresponding to the account that you want to enable.

10. Select **Enable account**. A confirmation dialog box is displayed confirming that account will be enabled and access will be granted or re-provisioned.

11. Select **Enable**. A success message is displayed.

    During the process, a transitional state indicator with an icon and tool tip are displayed. Once the process is complete, the final status changes to **Enabled** in the **Status** column.

Delete an Account Managed by Oracle Access Governance

You can directly delete one or more accounts that are managed by Oracle Access Governance. Once deleted, you can no longer manage these accounts from Oracle Access Governance.

Delete an Account Managed by Oracle Access Governance

1. Log in to Oracle Access Governance Console.

2. Click the navigation icon in the top left corner to display the navigation menu.

3. Select **Service Administration**, and then **Manage Identities**. The **Identities** page opens by default.

4. From the Identities list, select the **Actions** icon and select **View details**.

   The Identity details page is displayed with the **Permissions** tab selected by default.

5. Select the **Accounts** tab.

6. Select the ••• **Actions** icon corresponding to the account that you want to delete.
7. Select **Delete account**. A confirmation dialog box is displayed confirming that account will be deleted and access will be removed.
8. Select **Delete**. A success message is displayed.

   During the process, a transitional state indicator with an ⚠ icon and tool tip are displayed. Once the process is complete, the account is removed from the list.

Manage Identity Attributes

Identity attributes refer to properties of an identity, such as name, location, job code, organization name, and so on. Once you integrate Authoritative source systems, internally, a composite identity profile is constructed that contains **Core** and **Custom** attributes within Oracle Access Governance.

Oracle Access Governance supports the following attribute types:

- **Core attributes**: Fixed standardized identity attributes as recognized by Oracle Access Governance schema, mandatory for performing access management and reviews operation.
- **Custom attributes**: Additional non-standard identity attributes defined based on business needs beyond a standard set. For example, you may have a core attribute *Location*, while additional custom attributes, such as *Area*, *City*, or *Zip Code* must be configured, to support your business needs.

This composite identity profile can contain identity attributes from various Authoritative sources that you have integrated. This composite identity profile acts as a source of truth for Oracle Access Governance to perform various governance and provisioning operations.

Overview

Oracle Access Governance automatically fetches core and custom identity attributes defined in an Orchestrated System. Details of attributes are automatically loaded into Oracle Access Governance when data is ingested from an Orchestrated System. You can use these attributes in Oracle Access Governance to perform various functions, such as running access review campaigns, choosing identities for identity collections, or applying attribute conditions to enable/disable the available identity data set.

If you have defined custom attributes in your Orchestrated System , after the initial data load, you can choose to refresh the Oracle Access Governance schema to load the latest custom attributes.

To understand this better, let's look at a couple of examples:

- While creating a campaign, a *Campaign Administrator* selects custom attributes - *Cost Center* and *Department ID* to further refine the campaign selection criteria to run access review campaigns.
- While creating an identity collection, an *Administrator* can apply membership rules using the core and custom attributes. For instance, to create a senior management list of employees for the *Accounting* organization, create an identity collection to include employees where the *Job Level* is Director and above, and the *Organization* is Accounting.

Requirements and Rules for Managing Identity Attributes

Identity attributes are governed by certain requirements and rules. Let's see a few of them:

- A custom attribute that is encrypted in your schema will not be available in Oracle Access Governance and won't show up on the **Identity Attributes** page.
- You can choose to use the custom attribute value directly or modify the value of the attribute by applying transformation rules. For example, concatenating employee number with first name to set a display name.
- You cannot edit the default feature selections for core attributes.
- If you change a nested attribute (*<parent>.<child>*), then a list of additional dependent attributes will be affected and displayed. For example, if you update the orchestrated system for the attribute *name.firstName*. To ensure data integrity, the surname of the identity should come from the same Orchestrated System, so a message will be displayed *This will also change the orchestrated system for attributes: name.lastName*. When you save the change, both attributes will be updated.
- For Oracle Cloud Infrastructure (OCI) orchestrated system, an additional option is displayed, **Which domain?**. If you have multiple domains in your OCI tenancy, select an appropriate domain to use as the source of truth for your identities. If you have already run a dataload from your OCI Orchestrated System, you can select from a list of available domains ingested from the OCI system. If the dataload has not been run you can enter the domain name using free text.

View Attributes

As an *Administrator*, you can view, and search for available core and custom identity attributes, and manage the enabled Oracle Access Governance features for these attributes.

Here's how you can view the available custom attributes:

1. In the Oracle Access Governance Console, from the navigation menu, select **Service Administration** , and then select **Identity Attributes**.
   The **Identity Attributes** page is displayed. You can view the available core and custom attributes, which are displayed on the **Core** and **Custom** tabs respectively.

View Attribute Details

You can view the following attribute details:

Field	Description
Attribute name	Original attribute name as available in the Orchestrated System that is connected with Oracle Access Governance.
Orchestrated system	Orchestrated system name from which the attribute is populated.
Display name	Unique attribute name that will be used within Oracle Access Governance Console for easy identification and usage.
Type	Data type of the attribute.

Flags	• **Identity details**: If selected, attributes are displayed in: – Who Has Access to What functionality where you can view resource details for an identity. – My Access Reviews functionality where you can perform access reviews and see review insights. You can select up to 250 attributes for this feature • **Campaign selection**: If selected, the attribute is available for use in user access review campaigns. • **Event-based Setup**: If selected, the attribute is available for use in configure event-based triggers for identity access reviews. • **Manage Identities**: If selected, the attribute is available for use in configure activation rules to manage identities from Oracle Access Governance, and to enable custom attributes in creating an identity collection.

Search and Filter Custom Attributes

Use the **Search** field to locate the required attribute by the attribute name. You can manage a large set of attributes by applying filters based on the suggested filters. For example, selecting **Identity details On** will display all the attributes for which the Identity details flag is enabled.

On the top-right side of the page, select an orchestrated system to see attributes specific to that orchestrated system. If you select **No system available**, then you'll see a list of attributes not associated with any active orchestrated system.

Manage Core Attribute Settings

You can modify core attribute settings in a number of ways, including updating the Orchestrated System from which the attribute is populated, and applying data transformation rules to modify the incoming attribute value. You cannot change the default feature selections for core attributes.

To modify core attribute settings perform the following steps on the **Identity Attributes** page:

1. In the Oracle Access Governance Console, from the navigation menu, select **Service Administration**.

2. From the **Core** tab, click the **Edit** icon corresponding to the core attribute that you want to modify.

   The identity attribute fields are displayed in the editable mode letting you update attributes in a single edit operation.

3. To update which Orchestrated System should be used to populate the attribute, select an appropriate Orchestrated System from the **Which orchestrated system?** list.

   For Oracle Cloud Infrastructure (OCI) orchestrated system, an additional option is displayed, **Which domain?**. If you have multiple domains in your OCI tenancy, select an appropriate OCI Identity and Access Management domain to use as the source of truth for your identities.

4. Use direct attribute value or add a rule to apply inbound data transformation rules:

   a. In **Populated**, select the **Change** link.

   b. For using the attribute value as-is with no data transformation, select **Use the <attributename> value directly**. This action displays the value **Directly** in the **Populated** field.

   c. For applying rules, select **Build a rule around the <attributename>**.

   d. Enter the rule and click **Validate** to check your syntax. For further details on syntax refer to Data Transformation for Inbound and Outbound Rules. You cannot apply rules to nested attributes (*<parent>.<child>*).

   e. Click **Apply**. This action displays the value **By rule** in the **Populated** field.

5. After performing your edits, click  **Apply**. This preserves your changes. You can continue editing other attributes following the same process. The **Last updated by** column for the attributes that have been updated will display **Modified**.

6. Click **Save** to apply your changes and update all the attributes at once.

   The **Last updated by** column displays the administrator name who performed the most recent update.

Manage Custom Attribute Settings

You can modify custom attribute settings in a number of ways, including updating the Orchestrated System from which the attribute is populated, modifying the display name, applying rules to perform data transformations on the inbound value, and including/excluding the use of the attribute for certain Oracle Access Governance features.

To modify custom attribute settings perform the following steps on the **Identity Attributes** page:

1. In the Oracle Access Governance Console, from the  navigation menu, select **Service Administration**.

2. From the **Custom** tab, click the  **Edit** icon corresponding to the custom attribute that you want to modify.

   The identity attribute fields are displayed in the editable mode letting you update all the attributes in a single edit operation.

3. To update the display name, in the **What is the display name?** field, set the display name for the attribute selected.

4. To update which Orchestrated System should be used to populate the attribute, select an appropriate Orchestrated System from the **Which orchestrated system?** list.

   For Oracle Cloud Infrastructure (OCI) orchestrated system, an additional option is displayed, **Which domain?**. If you have multiple domains in your OCI tenancy, select an

appropriate OCI Identity and Access Management domain to use as the source of truth for your identities.

5. Use direct attribute value or add a rule to apply inbound data transformation rules:
   a. Select the **Change** link.
   b. For using the attribute value as-is with no data transformation, select **Use the <attributename> value directly**.
   c. For applying rules, select **Build a rule around the <attributename>**.
   d. Enter the rule and click **Validate** to check your syntax. For further details on syntax refer to Data Transformation for Inbound and Outbound Rules. You cannot apply rules to nested attributes (*<parent>.<child>*).
6. Select or Clear the appropriate feature check box to include or exclude the attribute from the feature. For example, include Cost center while setting up campaigns.
7. After performing your edits, click  **Apply**. This preserves your changes. You can continue editing other attributes following the same process. The **Last updated by** column for the attributes that have been updated will display **Modified**.
8. Click **Save** to apply your changes and update all the attributes at once.

   The **Last updated by** column displays the administrator name who performed the most recent update.

Fetch Latest Custom Attributes

If you don't see the latest custom attributes in the list, click the **Fetch attributes** button.

This action will run the schema discovery on the orchestrated system, and fetch the latest schema objects to get the updated list of custom attributes. If new custom attributes are available, then the schema discovery process may take a couple of minutes to complete, and show the updated list of custom attributes.

> **Note:**
>
> If you have an encrypted attribute in your schema, then this process won't fetch and show up that encrypted attribute on this page.

Whenever a new custom attribute is added, you first need to enable that attribute for the features where you want to use it.

> **Note:**
>
> This action won't ingest the attribute data from the orchestrated system but will just load the schema objects. To fetch and use the attributes' data, you either have to wait for the next upcoming scheduled data sync operation or manually run the data load operation. See the Configure Settings for an Orchestrated System topic.

Match Unmatched Accounts to an Identity

Unmatched accounts are those accounts from a Managed System which do not match any identities in Oracle Access Governance. Unmatched accounts can be managed by administrators in the Oracle Access Governance Console.

You can match unmatched accounts to an identity in Oracle Access Governance as described in the following:

1. In your browser, navigate to the Oracle Access Governance service home page, and login as a user with the *Administrator* application role.
2. Navigate to the **Unmatched Accounts** page by one of the following methods:
   a. Select the **Service Administration** tab from the Oracle Access Governance service home page, and click **Select** on the **What accounts exist that are not assigned to an identity?** tile.
   b. On the Oracle Access Governance service home page, click on the  icon, then select **Service Administration** → **Unmatched Accounts**.
3. On the **Unmatched Accounts** page, a list of all unmatched accounts is displayed, with the following information:
   - **Account**: the account name
   - **Target**: the target Managed System the account was ingested from.
   - **Date created**: the date the account was created on the target Managed System.
   - **Insights**: insights into why the account is unmatched. These include:
     – **Multi-match**: account matches multiple identities in Oracle Access Governance
     – **No match**: account matches no accounts in Oracle Access Governance
     – **User deleted**: account matches a identity in Oracle Access Governance which has been deleted in a source Managed System, but which still has accounts open on other Managed Systems.
4. You can filter the list of unmatched accounts displayed in the following ways:
   a. Search using free text.
   b. Select a specific insight to filter on.
   c. Select a target from the **Target** drop down list to filter on.
5. You can download a CSV file of the unmatched accounts by selecting the CSV download icon, 

. Selecting this download will save the unmatched accounts according to any filters which have been applied. The results are saved to a file, `unmatchedAccounts-report.csv`.
6. Once you have identified an account you want to match to an existing Oracle Access Governance identity, you can manage it using the following steps:
   a. Select the actions menu, for the account you wish to match.
   b. Select the **Match to identity** action.

c. The **All identities** tab will display for insight types/ Search for the identity you want to match the account to.

d. Select the identity you want to match, and click **Match to identity**.

e. If the insight type is **Multi-match** then an additional tab is displayed for **Suggested identities**. This provides suggestions for the identities you can match to your unmatched account. You can accept one of the suggestions, or navigate to the **All identities** tab and search as detailed in the previous steps.

Remove Unmatched Account

In certain circumstances you may not be able to match an unmatched account with an identity. In this case, you may want to remove the orphan account from Oracle Access Governance and the Managed System it was ingested from. This prevents the unmatched account from being re-ingested in subsequent dataloads, or used in any other way in your environment.

To remove an unmatched account from your environment:

1. In your browser, navigate to the Oracle Access Governance service home page, and login as a user with the *Administrator* application role.
2. Navigate to the **Unmatched Accounts** page by one of the following methods:
   a. Select the **Service Administration** tab from the Oracle Access Governance service home page, and click **Select** on the **What accounts exist that are not assigned to an identity?** tile.
   b. On the Oracle Access Governance service home page, click on the  icon, then select **Service Administration** → **Unmatched Accounts**.
3. Identify the unmatched account you want to remove.

   a. Select the actions menu, for the account you wish to remove.
   b. Select the **Remove** action.

   The unmatched account is removed from your environment. This includes removing it from Oracle Access Governance so it will no longer be displayed as an unmatched account. The provisioning feature of Oracle Access Governance will also propagate the removal back to the Managed System from which the account was ingested, removing the account from here as well.

Normalise Key-Values Across Orchestrated Systems

Global Key-Values allow you to normalise attribute values across orchestrated systems by providing key-value pairs that can be utilized by in-bound and out-bound transformations, and in account attributes. This allows you to standardize attribute values across your enterprise for items such as country code.

Global Key-Values is a feature which allows you to load a CSV file containing key/value pairs and make them available to:

- In-bound transformations

- Out-bound transformations
- Account attributes

For example, you might have a requirement to standardize country codes according to the ISO alpha-2 and ISO alpha-3 standards. Global Key-Values allow you to set a lookup for each of these 2 and 3 digit codes together with the corresponding country name (e.g. *US*, *USA*, *United States of America*). When populating an account attribute or setting values via an in-bound or out-bound transformation, you can utilize these Global Key-Values to maintain consistency across your enterprise.

Create a Global Key-Value

Create a Global Key-Value by uploading a CSV file with key/value pairs into Oracle Access Governance

To create a Global Key-Value, perform the following steps:

1. In your browser, navigate to the Oracle Access Governance service home page, and login as a user with the *Administrator* application role.
2. Navigate to the **Global Key-Values** page by clicking on the  icon, then select **Service Administration → Global Key-Values**. The page will display any existing Global Key-Values that have been setup previously.
3. Select **Create global key-values** to add a new global key-value definition.
4. In the **Create global key-values** pop-up enter the following values for the Global Key-Value required.
   a. **Name**: The name of the Global Key-Value you are creating, for example *CountryCodeISO2*.
   b. **Description**: A description for the Global Key-Value you are creating, for example *ISO alpha-2 country codes*
   c. **Select a file**: Select a CSV file containing the key/value pairs you want to create for this Global Key-Value. An example for the ISO2 example might include:

```
UA,Ukraine
AE,United Arab Emirates (the)
GB,United Kingdom of Great Britain and Northern Ireland (the)
UM,United States Minor Outlying Islands (the)
US,United States of America (the)
```

      The CSV file you upload should comply with the following requirements:

      - Files should be UTF-8 encoded
      - Keys must be unique
      - The string length of keys and values must both be less than 256 characters
      - Key/value pairs cannot exceed 10k

   d. Click **Create** to save the Global Key-Value.

View Details of a Global Key-Value

You can view the details of your Global Key-Value using the Oracle Access Governance Console.

To view details of a Global Key-Value, perform the following steps:

1. In your browser, navigate to the Oracle Access Governance service home page, and login as a user with the *Administrator* application role.
2. Navigate to the **Global Key-Values** page by clicking on the icon, then select **Service Administration → Global Key-Values**. The page will display any existing Global Key-Values.
3. Select the Global Key-Value that you want to manage and select **View Details**.
4. Details of the status and values assigned to the Global Key-Value are displayed.
5. You have the option to export the key/value pairs by selecting the 

button.

Edit Details of a Global Key-Value

Edit details of your Global Key-Value using the Oracle Access Governance Console.

To edit a Global Key-Value, perform the following steps:

1. In your browser, navigate to the Oracle Access Governance service home page, and login as a user with the *Administrator* application role.
2. Navigate to the **Global Key-Values** page by clicking on the icon, then select **Service Administration → Global Key-Values**. The page will display any existing Global Key-Values.
3. Select the Global Key-Value that you want to manage and select **Edit**.
4. In the **Edit global key-values** pop-up amend the following values for the Global Key-Value as required.
   a. **Name**: The name of the Global Key-Value you are creating, for example *CountryCodeISO2*.
   b. **Description**: A description for the Global Key-Value you are creating, for example *ISO alpha-2 country codes*
   c. **Select a file**: Select a CSV file containing the key/value pairs you want to create for this Global Key-Value. You might want to amend the CSV file to add updates to the key/value pairs, or you may need to reload a CSV file when you have had a failure during creation of the Global Key-Value and the status is set to *Failed*.
   d. Make your changes and click **Update** to save the Global Key-Value.

Delete a Global Key-Value

Delete a Global Key-Value using the Oracle Access Governance Console.

To delete a Global Key-Value, perform the following steps:

1. In your browser, navigate to the Oracle Access Governance service home page, and login as a user with the *Administrator* application role.
2. Navigate to the **Global Key-Values** page by clicking on the icon, then select **Service Administration → Global Key-Values**. The page will display any existing Global Key-Values.

3. Select the Global Key-Value that you want to delete and click on the actions menu. Select **Delete**.
4. Confirm your deletion in the pop-up by selecting **Delete**

> **Note:**
>
> Reference checking is not supported with Global Key-Values. If you delete a Global Key-Value that is referenced in an account attribute or transformation then the lookup values provided by it will return null values.

4

Access Controls

Account Lifecycle Management - Automated Provisioning for Joiners Movers and Leavers (JML) Process

Oracle Access Governance supports automated provisioning and de-provisioning of accounts and accesses based on the identity lifecycle stage. Identity Lifecycle involves three key stages - Joiners, Movers, and Leavers, popularly known as the JML process. Support for this process involves creation, modification, and deletion of identity accounts and their access permissions based on attribute change in the integrated Orchestrated system.

This process ensures that identities get the required access automatically without raising the access request manually. It not only reduces the administrative burden but also ensures data integrity and compliance. Other ways of provisioning are to request the access manually or directly provision it from the Managed System. For more information, see View My Access Requests.

As a user with `AG_ServiceDesk_Admin` role, you can directly manage account lifecycle without any approval workflows. From the **Manage Identities → Identities** page , you can enable, disable, delete accounts, or terminate all accounts and associated accesses for an identity. You can also retry provisioning for failed or pending statuses, and revoke permissions assigned directly or through requests. For detailed steps, see Manage Account Lifecycle with Service Desk Executive Support.

With Oracle Access Governance:

- Joiners get their birth-right access when they join the enterprise.
- Movers get the necessary accesses when they change roles, get internal transfers, or get promotions within the enterprise.
- Leavers have their account revoked (delete or disabled) once they exit the enterprise.

In Oracle Access Governance, your identity information is built up using a set of **Core** and **Custom** identity attributes. Whenever you create, modify, or update an identity record in the Authoritative Source, Oracle Access Governance ingests the latest data in the upcoming data load operation, and initiates the corresponding provisioning/de-provisioning operations. Oracle Access Governance achieves this granular and flexible access control mechanism by using the *Policy-Based Access Control* (PBAC) model. Oracle Access Governance assigns membership to identities using the attributes (*Attribute-Based Access Control (ABAC)*), and then provisioning the identities based on defined policies. A policy may further leverage the *Role-Based Access Control* model to assign appropriate role-based permissions ingested from identity attributes.

Supported Operations: Create Account, Read Account, Assign Permissions, Revoke Permissions, Change Password, Disable Account, Update Account, Delete Account. For additional details, refer to the specific Orchestrated System documentation as mentioned in Supported Integrations in Oracle Access Governance.

Employees Onboarding - Joiners Provisioning

When a new employee joins or gets hired in an enterprise, a new record gets created in the Authoritative Source, such as Oracle HCM. Once identities are onboarded in Oracle Access Governance, birth-right access or default set of accounts and permissions can be provisioned, based on the Access Control configurations done in Oracle Access Governance.

Joiners process ensures that every new employee gets the necessary account and permissions to start-off their onboarding process.

When an identity gets onboarded and is *Active* in Oracle Access Governance, all the identity attributes are compared against the defined policies. If an Oracle Access Governance policy grants certain Role or Access Bundle access to identities belonging to a specific department, then they are provisioned for that role or Access Bundle.

Scenario: When a new employee, *Alice*, joins the *Customer Success* department of the *Sales* division, *Joiners* provisioning ensures Alice receives all the mandatory accounts and permissions applicable to her division and her department. Let's look at how to achieve this in Oracle Access Governance.

Executing Joiners Provisioning in Oracle Access Governance

Taking the above scenario, let's look at the high-level steps involved to achieve *Joiners* provisioning in Oracle Access Governance:

1. As an *Access Control Administrator*, set up the Access Control configuration, as follows:
    a. Create an **Identity Collection** based on membership rules. For example, create an Identity Collection with membership rule as *Source Organization* equals *Sales* and another Identity Collection where *Department* equals *Customer Success*. For more details, refer to Create Identity Collections.
    b. Create an **Access Bundle** or **Role**, and package access to necessary permissions. For example, create an Access Bundle *Sales_AB* with permissions applicable to *Sales* and another Access Bundle *Customer_Success_AB* with permissions applicable to *Customer Success*. For more details, refer to Create Access Bundle.
    c. Create a Policy and associate the permissions part of the Access Bundle with Identity Collection. For example, create a Policy *Sales_Policy* and associate *Sales_AB* with *Sales* Identity Collection. Similarly, create *Customer_Success_Policy* and associate *Customer_Success_AB* with *Customer Success* Identity Collection. For more details, refer to Create a Policy.
2. **Authoritative Source** registers a new record of an employee. For example, HR adds a new record of Alice with *Business Unit* as *Sales* and *Department* as *Customer Success*.
3. **Orchestrated System** performs data load, ingests latest data and builds composite identity profile in Oracle Access Governance. For more information, refer to Identity Orchestration Process Flow.

A new identity profile gets created in Oracle Access Governance. The attributes are matched against the defined policies and appropriate provisioning operations are triggered. For *Joiners*, Orchestrated System triggers **Create Account** and **Add account or permission data** provisioning operations to assign new accounts and permissions.

Validate Joiners Provisioning in Oracle Access Governance

- As an *Enterprise-wide Access Administrator*, you can search identity to view complete identity details displaying identity attributes, permissions, account information. You can also view identity collection details to verify the new member list.

- As an Identity Manager, you can see comprehensive identity details for the direct reports in the **Who has Access to What → My Directs' Access**.
- As a *User*, you can validate your accounts and permissions from the **My Stuff → My Access** page.

Depending on the **Account settings** configured for the Orchestrated system, a *User* or *User manager* will receive notification whenever new accounts are created. By default, the notifications are sent to *User*. For more information, refer to Configure Orchestrated System Account Settings.

Employee Transfers - Movers Provisioning

When an employee internally transfers, relocates, or gets promotion within an organization, a record gets updated for that employee in the Authoritative Source. Upon transferring, identity should only have access to suitable privileges relevant to the new job profile. Remaining accounts and permissions should be revoked based on the Account Lifecycle settings. You can achieve this automatic provisioning based on the Access Control configurations done in Oracle Access Governance.

Movers process ensures that only the necessary and correct set of permissions or accounts are assigned to the employees that they require in their new role.

Scenario: When an employee, *Alice*, gets internal transfer from the *Customer Success* department to the *Cloud Sales* department of the *Sales* division, *Movers* provisioning ensures Alice receives all the privileges applicable to her new role, and revokes or disables prior accounts and permissions needed by her former role. In this example, Alice will continue to have permissions applicable for *Sales* division but will get new privileges relevant in the *Cloud Sales* department. If no longer applicable, her prior accounts gets either disabled or revoked, and permissions associated with the accounts are also removed. Let's look at how to achieve this in Oracle Access Governance.

Executing Movers Provisioning in Oracle Access Governance

Taking the above scenario, let's look at the high-level steps involved to achieve *Movers* provisioning in Oracle Access Governance:

1. As an *Access Control Administrator*, you must have this minimum set-up, as follows:
    a. Create an **Identity Collection** based on membership rules. For example, create an Identity Collection with membership rule as *Department* equals *Cloud Sales* and another Identity Collection where *Department* equals *Customer Success*. For more details, refer to Create Identity Collections.
    b. Create an **Access Bundle** or **Role**, and package access to necessary permissions. For example, create an Access Bundle *Cloud_Sales_AB* with permissions applicable to *Cloud Sales* and another Access Bundle *Customer_Success_AB* with permissions applicable to *Customer Success*. For more details, refer to Create Access Bundle.
    c. Create a Policy and associate the permissions part of the Access Bundle with Identity Collection. For example, create a Policy *Cloud_Sales_Policy* and associate *Cloud_Sales_AB* with *Cloud Sales*. For more details, refer to Create a Policy.
2. **Authoritative Source** records an update for the identity. For example, HR updates Alice's department from *Customer Success* to *Cloud Sales*.
3. **Orchestrated System** performs data load, ingests latest data and builds composite identity profile in Oracle Access Governance. Based on your account lifecycle settings for an orchestrated system, permissions or accounts are either disabled or revoked. For more information, refer to Identity Orchestration Process Flow.

Users with the `AG_ServiceDesk_Admin` role can directly revoke permissions from the **Manage Identities** page, using the **Revoke permission** operation. The Grant Type of these permissions must either be `DIRECT` or Access Bundles granted through `REQUEST`. You cannot revoke permissions for Oracle Cloud Infrastructure (OCI) or Oracle Identity Governance (OIG) systems. For detailed steps, see Revoke one or multiple permissions for an Account.

Validate Movers Provisioning in Oracle Access Governance

For *Movers*, Orchestrated System typically triggers the following operations:

- To disassociate former permissions with the identity accounts, it triggers **Remove account or permission data**.
- To disable the accounts, it triggers **Update Account**, or to delete the accounts, it triggers **Revoke** .
- To associate new accounts and permissions, it triggers **Create Account** and **Add account or permission data**.
- If only permissions are different, then Account remains enabled but **Add account or permission data** and/or **Remove account or permission data** operations are triggered to update the permissions for that account.
- If a disabled account is enabled, then it triggers **Update Account** along with **Add account or permission data** and/or **Remove account or permission data**.

You can verify the changes on the Oracle Access Governance Console:

- As an *Enterprise-wide Access Administrator*, you can search identity and view complete identity details displaying identity attributes, permissions, account information. You can also view identity collection details to verify the new member list.
- As a *User*, you can validate your accounts and permissions from the **My Stuff → My Access** page.

Depending on the **Account settings** configured for the Orchestrated system, a *User* or *User manager* will receive notification whenever new accounts are created. By default, the notifications are sent to *User*. The existing accounts can either be deleted or disabled depending on the Account Settings. For more information, refer to Configure Orchestrated System Account Settings.

Employees Offboarding - Leavers De-Provisioning

When an employee exits the enterprise, a record gets deleted or disabled in the Authoritative Source. Upon exiting, all the accounts and associated privileges assigned to that identity will either be deleted or disabled from the Managed System.

Leavers process ensures that all accounts and permissions assigned to the identity are automatically revoked upon their exit. When an identity exits, and is marked *Inactive* in Oracle Access Governance, identity accesses are either revoked or disabled based on account settings.

Users with the `AG_ServiceDesk_Admin` role can directly revoke permissions from the **Manage Identities** page, using the **Revoke permission** operation. The Grant Type of these permissions must either be `DIRECT` or Access Bundles granted through `REQUEST`. You cannot revoke permissions for Oracle Cloud Infrastructure (OCI) or Oracle Identity Governance (OIG) systems. For detailed steps, see Revoke one or multiple permissions for an Account.

Users with the `AG_ServiceDesk_Admin` role can now directly disable accounts managed by Oracle Access Governance from the **Manage Identities** page, using the **Disable account**

operation. Once disabled all the associated accesses are revoked. The accounts can still be managed by Oracle Access Governance. For detailed steps, see Disable and Enable an Account Managed by Oracle Access Governance.

Scenario: When an employee, *Alice*, exits the enterprise, *Leavers* de-provisioning ensures all the assigned accounts and permissions applicable to her role gets revoked (delete or disabled). Let's look at how to achieve this in Oracle Access Governance.

Executing Leavers De-Provisioning in Oracle Access Governance

Taking the above scenario, let's look at the high-level steps involved to achieve *Leavers* de-provisioning in Oracle Access Governance:

1. As an *Access Control Administrator*, you must have this minimum set-up, as follows:
    a. An **Identity Collection** based on membership rules. For more details, refer to Create Identity Collections
    b. An **Access Bundle** or **Role** where necessary permissions are packaged together. For more details, refer to Create Access Bundle and Manage Roles.
    c. A Policy that associates the permissions (through Access Bundle) with Identity Collection. For more details, refer to Create a Policy.
2. **Authoritative Source** deactivates an existing record of an employee in the system.
3. **Orchestrated System** performs data load, ingests latest data. For more information, refer to Identity Orchestration Process Flow.

When an identity profile is deactivated and the data load is successful, a **Revoke** or **Update Account** provisioning task is triggered to either delete or disable the identity's accounts. Permissions associated with the account gets revoked and **Remove account or permission data** is triggered to remove permissions from the Managed system. For more information, refer to Configure Orchestrated System Account Settings.

Identity Collections are groups of identities based on shared attributes or named identities. Identity collections comprise identities that have been on boarded from integrated systems using identity orchestration.

Identity Collections simplify tasks by allowing you to configure features for a collection of identities, rather than for each individual identity. You can use Identity Collections to

- Associate identities with appropriate access bundles or roles using policies.
- Delegate Access Review tasks to an Identity Collection.
- Assign as approvers in Approval Workflows.

Navigate to Identity Collections

Here's how you can access the Identity Collections page:

1. Sign in to the Oracle Access Governance Console .
2. Click the  icon, and select **Access Controls** and then **Identity Collections**. You will see the **Identity Collections** page where you can view and manage existing identity collections, or create new ones.
3. To create a new identity collection, click the **Create an identity collection** button.

The **Create a new identity collection** page is displayed.

Add Details

In the **Add Details** task, you can enter specifics about your identity collection. Here, you can give a meaningful name to your identity collection and add its supporting description.

> **Note:**
>
> By default, all identities enabled in the Licence Management service, can utlize all identity data attributes, including custom attributes, to create identity collections.

1. Enter a name for your identity collection in the **What do you want to call this identity collection?** field.
2. Add a description for your identity collection in the **How would you describe this collection?** field.
3. Select one or more identities from the **Who can manage this identity collection** list. The owner along with the listed identities can manage this identity collection.
4. Add one or more tags to identify or search your identity collection.
5. Once you have set your preferences, select **Next** to go to the *Select Identities* step.
6. Optional: You may click **Cancel** to cancel the current process.

Add Primary and Additional Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

No special application roles are necessary for assigning resource ownership. Any Oracle Access Governance active user can be assigned as the owner of the resources. All the owners can read, update, or delete the resources that they own. However, the *Primary Owner* is assigned as the access reviewer when you choose the **Owner** template in the approval workflow for performing *Ownership reviews* in Campaigns. For more information, refer Types of Access Reviews Offered by Oracle Access Governance.

For assigning resource ownership, you must have active Oracle Access Governance users. When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

   You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Add system

The **Add system** task will display if you have at least one orchestrated system, in your enterprise, where you have selected to manage identities. This option allows you to select

whether the identity collection you are creating will manage a group on the orchestrated system or not.

1. Select **Yes** or **No** in response to the question **Will this identity collection manage a group on a system?** field.
2. If your answer is **No** then this identity collection will not manage group for the system and you can select **Next** to go to the *Select Identities* step. If your answer is **Yes**, then this identity collection will manage a group on a system, and you should complete the steps which follow.
3. On selecting **Yes**, you will be prompted to identify a system for which the identity collection will manage a group. Select **Add system**. In the **Add system** panel, select the system name from the drop-down list. The selected (new or existing) group in this system will be managed by this identity collection.
4. Complete the group details for the system. The details required will depend on the system type selected.

   Check the **Manage existing group** check box if you want to create the identity collection from an existing group in the system. If you select this option then there is no requirement to enter the group details.
5. Complete the account details for the system. The details required will depend on the system type selected.
6. Click **Save** to save the group details of the system.
7. Click **Next** to progress to identity selection, **Save draft** to save this as a draft identity collection, or **Cancel** to cancel the current process.

Select Identities

In the **Select Identities** task, you have to select identities for your identity collection.

You can select identities based on:

- **Membership rule**: Set criteria based on certain conditional statements. Either one (**Any**) or all (**All**) the set conditions must be satisfied. The list of available attributes is determined by the data ingested from the orchestrated systems.
- **Named identities**: Search and select one or more users by their full name that you want to include in your identity collection. The list of available users is determined by the data ingested from the orchestrated systems.
- **Both Membership rule and Named identities**: You can have a combination of both membership rule and named identities to set criteria for your identity collection.

> **Note:**
>
> You can also exclude specific members from your identity collection, by selecting **Manage exclusions** and entering the identities you want to exclude.

Add Identities based on Membership Rule

To add identities based on conditional statements, select the **Membership rule** tab.

The identities satisfying the set criteria will automatically be included in that identity collection. For example, for an identity collection, if you set the conditional rule to **Department** *Equals*

Finance, then all the human identities belonging to the Finance department will be included in that identity collection.
To set the conditional rule for identities, do the following:

1. Select **Any** if any one of the set conditions should be satisfied, or select **All** if all the set conditions must be satisfied for that identity.
2. Select the attribute name from the list

> **Note:**
>
> Based on the orchestrated systems, you can select both core and/or custom attributes. To enable custom attributes, see View and Configure Custom Identity Attributes

3. Select the conditional operator. Based on the data type of the attribute selected, the usage of these operators will vary.
4. Type the attribute value.
5. Continue to add the conditional statements or rules for more attributes.
   By default all the identities matching the criteria will be included.
6. However, you can exclude certain identities from your conditional statements.

   Click the **Manage Exclusions** button next to **Excluding # identity from the attribute conditions** and then select the identities that you want to exclude from the identity collection.
   As you set the conditions or add identities, you can see the effect on the right-side of the screen of which identities are excluded and the applied membership rule.
7. Once you have set your preferences, select **Next** to go to the *Review and submit* step. You may select one of the additional actions:
   - **Save as draft**: To save your changes and later come back and edit the identities.
   - **Cancel**: To cancel the current process.
   - **Back**: To go back to the previous step.

Add Identities based on Named Identities

To directly add identities based on their full name, select the **Included named identities** tab.

All the available active identities (configured from the Licence Management page) will be displayed. In the user tile, you can view user details, such as full name, email address, organization name. Search or select one or more user tile that you want to include in your identity collection. As you select the identities, you can see the effect on the right-side of the screen of which identities are included. Once you have set your preferences, select **Next** to go to the *Review and submit* step.

Review and Submit

The Review and Submit step displays the information you have added in the previous steps.

You can see the preview of your identity collection. For this, click the **preview the identity collection** link available on the right-side of the page. If you are satisfied with your identity collection preview, click **Create**. You may select addition actions:

- **Save as draft**: to save your changes and edit the identity collection later.

- **Cancel**: To cancel the process.
- **Back**: To go back to the previous step.

Oracle Access Governance users can view and manage identity collections from the Oracle Access Governance Console.

View and Manage Identity Collections

You can view existing identity collections and manage the ones that you created, or are authorized to manage, using the Oracle Access Governance Console.

Follow the steps to navigate to the **Identity Collections** page:

1. Sign in to the Oracle Access Governance Console with a user assigned with the *Access Control Administrator* application role.
2. Click the icon, and select **Access Controls** and then **Identity Collections**. You will see the **Identity Collections** page where you can view and manage existing identity collections.

Here, you can see the count of existing identity collections and see the identity collections summary in the grid format, that includes:

- **Name**: Identity collection name.
- **Status**: **Active** or **Draft**
- **Owner**: Name of the owner who created this identity collection.
- **Targets**: the name of the target for which this identity collection manages groups.
- **Last updated**: Date on which the identity collection was last modified.
- **Tags**: User-defined tags for quick search and easy identification of the identity collection.

Use the *Actions* menu icon to **Edit**, **Delete** or **View Details** of the identity collection.

> **Note:**
>
> Users who own the identity collection or authorized users (selected while creating/ modifying an identity collection) can edit or delete the identity collection.

Search and Filter Identity Collections

You can use the **Search** field to locate the required identity collection by its name. You can narrow down the results by applying filters:

- **Updated Last Month**: You can view all the identity collections that were updated within the last month.
- **Updated Last Week**: You can view all the identity collections that were updated within the last week.
- **Created by Me**: You can view the identity collections that you have created.

- **Status Draft**: You can view the identity collections that haven't been created and are in the draft state.
- **Status Active**: You can view the identity collections that have been created and can be used within Oracle Access Governance.

You can limit the scope of the identity collections displayed by selecting the scope list of values, located in the top right of the Identity Collections page. The default value for scope is **All** which displays all Identity Collections. Select **Managed for systems** to restrict the scope of the identity collections displayed to those which manage groups for a system.

Edit an Identity Collection

The **Edit an Identity Collection** page provides the same guided tasks as you see while creating a new identity collection.

Owner of the identity collection and/or authorized users can modify its description, identity type, or added identities.
To do so:

- Click the ••• *Actions* menu icon corresponding to the identity collection that you want to modify, and then select **Edit**.

After updating your identity collection details, on the *Review and submit* step, select **Update** to update the identity collection. Alternatively you can select **Back** to edit values, or select **Cancel** to discard your changes.

View Details for an Identity Collection

To view details of an identity collection:

- Click the ••• *Actions* menu icon corresponding to the identity collection that you want to view, and then select **View Details**. You will see the *Identity Collections* page displaying the details, such as *Identities*, *Identity Type*, *Created by*, and *Last updated*.

On the left panel, you will see the following details:

- In the donut chart, you can see the total count of identities, and break up of identities in the collection that are included either through **Membership rule** or directly through **Included named identities**.
- If you have created an identity collection through **Membership rule**, you can see all the rules that helped to create the collection.
- If you have excluded an identity in the collection, you can see a list of excluded member(s).
- If you have created an identity collection directly using named identities, you can see a list included member(s).

On the rest of the page, you will see the identity names and their details in a tile format. You can search an identity using identity name.

Use the *Actions* menu to either **Edit** or **Delete** the identity collection. Owner of the identity collection and/or authorized users can edit or delete the identity collection.

Delete an Identity Collection

You can delete an identity collection as long as it is not associated with any delegation. If you are the owner of the identity collection or you have been given the rights by the identity collection creator, then you can delete the identity collection.

1. Click the *Actions* menu icon corresponding to the identity collection that you want to delete, and then select **Delete**.
2. On the confirmation pop-up, click **Delete** to remove the identity collection or click **Cancel** to retain the identity collection.

Policies are the mechanism by which you can provide resource access to identities within your organization. Policies associate resources and permissions with identities by means of roles and access bundles. Here we will see how you can maintain policies within your Oracle Access Governance service.

Create a Policy

You can create a policy in the Oracle Access Governance Console by following the steps below:

1. In your browser, navigate to the Oracle Access Governance service home page, and log in as a user with the *Administrator* or *Access Control Administrator* application role.
2. On the Oracle Access Governance service home page, click on the icon, then select **Access Controls → Policies**. This will take you to the **Policies** page, where you can view, edit, and create policies.
3. Click on the **Create a policy** button. This will take you to the **Create a new policy** flow, which guides you through the steps to setup a policy.
4. The **Let's get started building your policy** step of the flow allows to you provide a name and description for your policy, and select what types of associations to add to the policy. Enter values for the following:
   a. **What do you want to call this policy?**: Add a name for your policy.
   b. **How would you describe this policy?**: Add a description for your policy.
   c. **Would you like to add any tags?**: Enter any tags for this role that you would like to be able to search on. Examples may include regulatory compliance standards such as *SOX*, *HIPPA*, *GDPR* and others.
   d. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
   e. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.
   You can view the **Primary Owner** in the view list. All the owners can view and manage the resources that they own.

   Click the **Add a new association** tile, and select one of the following association types to add.

- **Access bundle association**
- **Role association**

5. When you have selected the association type to add, you will navigate either the **Add an access bundle policy assignment** or **Add a role policy assignment** flow, which guide you through the steps to assign your access bundle or role to a policy. The steps in the flow are mostly the same, where differences occur these will be highlighted in the instructions which follow.
6. **Select who** is the first step of the flow, where you add the identity collections to which the policy will apply. Select, or search for, the identity collections you want to include in your policy, and select **Next** to continue.
7. Depending on whether you selected access bundle or role association, the next step will be either **Select access bundles** or **Select roles**. Select the access bundle or role that you want to assign to the policy, and select **Next** to continue.
8. **Review and submit** is the next step. If you are happy with the information you have added, select **Add association** to save the association. You can also select **Back** to revisit the details you entered, or **Cancel** to abandon your changes.
9. If you add the association you will be taken back to the beginning of the **Create a new policy** flow. At this point you can elect to add another association to your policy by selecting **Add a new association** and adding the association as previously described. Once you are finished entering associations for the policy, select **Create** to save the policy, or **Save as draft** to save your policy for editing at a later date.

Edit a Policy

To edit an existing or draft policy, perform the steps described below.

1. On the Oracle Access Governance service home page, click on the icon, then select **Access Controls → Policies**. You can select the option to edit a policy in any one of the following ways:
   a. Select the name of the policy to navigate to the *View details* page. Click on the **Actions** menu and select **Edit**.
   b. From the list of policies, select the ••• *Action* menu. Select **Edit**.
   c. From the list of policies, select the ••• *Action* menu. Select **View details**. From the *View details* page, select **Edit**.
2. You navigate to the Policies workflow. Make any amendments and save your changes.

Delete a Policy

You can delete a policy using the Oracle Access Governance Console.

1. On the Oracle Access Governance service home page, click on the icon, then select **Access Controls → Policies** to navigate to the **Policies** page.

2. Select the name of the policy you want to delete, click on the **Actions** menu and select **Delete**.
3. You are prompted to confirm that you want to delete the policy. Select **Delete** to remove the policy, or **Cancel** if you decide to retain the policy.

A role is a group of access bundles. The access bundles contained within a role can span multiple targets. An example might be a role of *Database Administrator*, which groups together the *DBAdmin_Oracle*. *DBAdmin_DB2*, and *DBAdmin_MySQL* access bundles. This allows you to create roles which collect together the relevant access bundles to be able to perform that role. These roles can then be associated with identities via policies.

> **Note:**
>
> a role does not provide access to a resource by default. Access is given to an identity when a role is assigned to that identity via a policy or self-service request.

Create a Role

You can create a role in the Oracle Access Governance Console by following the steps below:

Create Role

1. In your browser, navigate to the Oracle Access Governance service home page, and log in as a user with the *Administrator* or *Access Control Administrator* application role.
2. On the Oracle Access Governance service home page, click on the  icon, then select **Access Controls → Roles → Create a Role**. Select **Create a role** from the Roles page. This will navigate you to the **Create a new role** flow, which guides you through the steps to setup a role.
3. **Role settings** is the first step of the flow. Enter values for the following:
   a. **Who can request this role?**: Define which identities can request this role. Select from one of the following values:
      - **No one**
      - **Anyone**
   b. **Which approval workflow?**: Select the name of the approval workflow you want to associate with this role from the list. If *No one* was selected in the previous step, then this selection will be disabled.
   c. **Would you like to add any tags to this resource?**: Enter any tags for this role that you would like to be able to search on. Examples may include regulatory compliance standards such as *SOX*, *HIPPA*, *GDPR* and others.
   d. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
   e. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.
   You can view the **Primary Owner** in the view list. All the owners can view and manage the resources that they own.

When you are happy with your selections, click **Next** to continue to the next step.

4. Next step is **Select access bundles**. You can search for, and select, access bundles that you want your role to contain. These access bundles can originate from multiple targets if required.
5. Next step is **Add details**. Enter values for the following:
   - **What is the name of this role?**: Enter a name for the role you are creating.
   - **How do you want to describe this role?**: Enter a description of the role you are creating.

   When you are happy with your inputs, select **Next** to proceed to the next step.
6. Next step is **Review and submit**. Here you will see a summary of the details you entered for the role, namely:
   - Name
   - Description
   - Requestable by
   - Approval workflow
   - Tags
   - Additional details

   If you decide not to create the role, select **Cancel** to reject the changes, else if you want to amend any details, select **Back**.

   If you are happy with the changes reviewed then you have three options to proceed.

   a. **Role assignment**: a role will not give access to anybody until it is assigned. If you select the **Start assignment after creation** checkbox, you will automatically navigate to the assignment flow on selecting the **Create role and assign** button. This is the default option.

   b. **Create role**: if you deselect the **Start assignment after creation** checkbox, you create the role, but do not navigate automatically to the assignment flow. On selecting the **Create** button, the role is saved and you are returned to the **Roles** page.

   c. **Save as draft**: you can select to save the role as a draft. You can edit the role later by selecting from the Roles page.

Assign Role

If you selected **Start assignment after creation**, or selected the *Add assignment* option, you will navigate to the role assignment page, which gives you the option to assign your role to an existing policy, or to create a new policy and assign the role to that. Initially, you are asked **Do you want to assign the role through an existing policy or create a new one?**

If you select **Existing policy**:

You can assign your role to an existing policy by following the steps below:

1. Select the policy you would like to add your role to from the drop-down list **Which policy do you want to assign it to?**.
2. You have the option **Do you want to add the role to existing associations or create a new one?**.
3. If the policy selected has existing role associations, then they are displayed under **Which associations do you want to add this role to?**. The role associations display as a tile which identifies which identities are associated with which roles. To add an association

between your newly created role and the identities, select the relevant role association, at which point your new role name is displayed on the tile. To save the association, click on **Add assignment**. Your role assignment is saved and you are returned to the **Roles** page.

If you select **Create a new policy**:

You create a new policy with the following steps:

1. Add the name of your policy in the **What do you want to call this policy?** field.
2. Add a description of your policy in the **How would you describe this policy?** field.
3. Under **Which identity collections do you want to associate this role with?** select the identity collections you want to associate with this role by selecting from the tiles displayed or entering a search.
4. Click on **Add assignment** to save your changes.

Edit a Role

To edit an existing or draft role, perform the steps described below.

1. On the Oracle Access Governance service home page, click on the icon, then select **Access Controls → Roles**. You can select the option to edit a role in any one of the following ways:
   a. Select the name of the role to navigate to the *View details* page. Click on the **Actions** menu and select **Edit**.
   b. From the list of roles, select the ••• *Action* menu. Select **Edit**.
   c. From the list of roles, select the ••• *Action* menu. Select **View details**. From the *View details* page, select **Edit**.
2. You navigate to the Role workflow. Make any amendments and save your changes.

Delete a Role

You can delete a role using the Oracle Access Governance Console.

1. On the Oracle Access Governance service home page, click on the icon, then select **Access Controls → Roles** to navigate to the **Role** page.
2. Select the name of the role you want to delete, click on the **Actions** menu and select **Delete**.
3. You are prompted to confirm that you want to delete the role. Select **Delete** to remove the role, or **Cancel** if you decide to retain the role.

Create an Access Bundle

An Access Bundle is a collection of permissions that package access to resources, application features, and functionality into a requestable unit. A specific access bundle will be associated with a single target.

Overview

With Access Bundles, you need not grant access to each permission individually but can request the access bundle for that resource. This simplifies the process of provisioning accounts with resource permissions.

ExampleApex Developer Access Oracle Access Governance

Manage Accesses using Oracle Access Governance Access Bundles

You can manage groups for Microsoft Entra ID (formerly Azure Active Directory) and Microsoft Active Directory.

For an Oracle Cloud Infrastructure (OCI) orchestrated system, for a particular domain, you can achieve:

- **Group Assignment**: Bundle OCI IAM groups in an access bundle, which can then be assigned to identities through a policy or an access request.
- **Application Role Assignment**: Bundle OCI cloud services application roles in an access bundle, which can then be assigned to identities through a policy or an access request.

Navigate to Access Bundle

To navigate to the Access Bundle page:

1. Sign in to the Oracle Access Governance Console with a user assigned either with the *Administrator* or *Access Control Administrator* application role.
2. You can select one of the following options to navigate to the Access Bundle page:
   - Click the navigation menu icon, and select **Access Controls**, and then **Access Bundles**.
   - On the console home page, click the **Access Controls** tab and then click the **Select** button on the **Manage Access Bundles** tile.

   Whichever option you choose, you will be navigated to the **Access Bundle** page, where you can create, view and manage access bundles.
3. To create a new access bundle, click the **Create an access bundle** button. The **Create a new access bundle** page is displayed.

Bundle Settings

In the *Bundle settings* task, you can enter general settings about your access bundle. You are also able to add user friendly tags that can be used in a search for this access bundle when creating policies.

1. Select the orchestrated system in the **Which system is this bundle for?** field.

You will see the applications available for selection, dependent on the data ingested from your integrated systems.

2. **[OCI-only]** Select domain in the **Which domain?** field from which you want to select application roles or OCI IAM groups.
3. **[OCI-only]** In the **Which type of permission?** field, select any one:
   - **Application role**: To package OCI application roles in an access bundle and assign it to identities.
   - **Group access**: To package and assign OCI IAM groups in an access bundle.

   You cannot combine **Application role** and **Group access** in a single access bundle. You may create a role in Oracle Access Governance and associate two separate access bundles with it. These can then either be requested through self service flows or provisioned through Oracle Access Governance policies. For details, see Manage Roles.
4. Select who can request this bundle from the available choices:
   - **Anyone**: Any identity can request the access to this access bundle.
   - **No one**: The access bundle can only be assigned by an Administrator through policies. You cannot request access to this bundle through self service flows.
5. Select the appropriate approval workflow in the **Which approval workflow should be used?** field.

   The displayed list is based on the custom approval workflows created in the Oracle Access Governance Console. For more information, see Create an Approval Workflow.

   > **Note:**
   >
   > If you have selected **No one** from the **who can request this bundle?** field, then **Which approval workflow should be used?** field will be disabled. Users won't be able to request the access bundle from the self-service module, but the Access Bundle can be provisioned using Policies.

6. Select one or more tags for this access bundle in the **Would you like to add any tags?** field. Examples might include **SOX**, **HIPPA**, **GDPR** or similar.
7. Once you are happy with your settings, click **Next** to go to the **Select permissions** task or click **Cancel** to cancel the current process.

Select Permissions

In the *Select Permissions* task, you can select permissions to include in this access bundle. Based on the orchestrated system, you may see additional attributes required for account provisioning. Refer to the specific orchestrated system articles to know more about the default attributes. For OCI, you can select OCI IAM groups or application roles.

1. Select one or more permissions associated with the target application. Alternatively, you can use the **Search** field to locate the required permission or role.
2. Once permissions are selected, click **Next** to go to the *Add Details* task.

Add Primary and Additional Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they

own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

No special application roles are necessary for assigning resource ownership. Any Oracle Access Governance active user can be assigned as the owner of the resources. All the owners can read, update, or delete the resources that they own. However, the *Primary Owner* is assigned as the access reviewer when you choose the **Owner** template in the approval workflow for performing *Ownership reviews* in Campaigns. For more information, refer Types of Access Reviews Offered by Oracle Access Governance.

For assigning resource ownership, you must have active Oracle Access Governance users. When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

   You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Add Details

In this *Add Details* task, you can give a name to your access bundle, and add a supporting description.

1. Enter name for your access bundle in the **What is the name of this bundle?** field.
2. Add a description for your access bundle in the **How do you want to describe this bundle?** field.

> **Note:**
>
> The other fields on the screen depends on the target type and permissions selected in the previous tasks.

3. Select the other values based on the selections made in the previous steps and click **Next** to go to the *Review and submit* task.

Review and Submit

The *Review and Submit* task displays the information you have added in the previous tasks.

If everything looks correct, then click **Create** to create the access bundle. You may select addition actions:

- **Cancel**: To cancel the process.
- **Back**: To go back to the previous step.
- **Save as draft**: To save the access bundle as a draft copy. This will display the access bundle on the **Access Bundle** screen with the status '**Draft**'.

Users with Oracle Access Governance Access Control Administrator rights can view and manage access bundle from the Oracle Access Governance Console.

View and Manage Access Bundles

Users with the Oracle Access Governance *Administrator* or *Access Control Administrator* application role, can view and manage access bundles from the Oracle Access Governance Console

Follow the steps to navigate to the **Access Bundle** page:

1. Sign in to the Oracle Access Governance Console with a user assigned with the Administrator application role.
2. Click the  icon, and select **Access Controls** and then select **Access Bundle**. The **Access Bundle** page is displayed where you can view and manage existing access bundles.

View Access Bundles

Here, you can see the count of existing access bundles and the details are listed in the grid that includes:

- **Name**: Access bundle name.
- **Status**: **Active** or **Draft**
- **Target**: Name of the target application.
- **Owner**: Name of the owner who created this access bundle.
- **Requestable by**: **Anyone**, or **No one**
- **Last updated**: Date on which the identity collection was last modified.
- **Tags**: User-defined tags for quick search and easy identification of access bundle.

Use the *Actions* menu icon to **View Details**, **Edit**, or **Delete** access bundles.

Edit an Access Bundle

The **Edit an Access Bundle** page provides the same guided tasks as you see while creating a new access bundle.

Owner of the access bundle can modify bundle settings, approval workflow, or selected permission.
To do so:

- Click the *Actions* menu icon corresponding to the identity collection that you want to modify, and then select **Edit**.

After updating your access bundle details, on the *Review and submit* step, select **Update** to update the access bundle. Alternatively you can select **Back** to edit values, or select **Cancel** to discard your changes.

View Details for an Access Bundle

- Click the *Actions* menu icon corresponding to the access bundle that you want to view, and then select **View Details**. You will see the Access Bundle details page displaying the description, what accesses are included for this access bundle , who can request this access bundle, what approval workflow will run to approve the request for this access bundle.
  You can click on the **Actions** button to edit or delete the access bundle.

Delete an Access Bundle

If you are the owner of the access bundle, then you can delete the access bundle.

1. Click the *Actions* menu icon corresponding to the access bundle that you want to delete, and then select **Delete**.
2. On the confirmation pop-up, click **Delete** to remove the access bundle or click **Cancel** to retain the access bundle.

Activating an Access Bundle

Clicking the **Save as draft** button while creating an access bundle saves the access bundle as a draft copy and will be displayed on the **Access Bundle** screen with the status **Draft**. If you are the owner of the access bundle, then you can activate such access bundle.

To do so:

1. Click the *Actions* menu icon corresponding to the access bundle that you want to delete, and then select **Activate**.
2. On the confirmation pop-up, click **Activate** to activate the access bundle or click **Cancel** to retain the status.

Oracle Access Governance users with the *Access Control Administrator* or *Administrator* application role can create approval workflows from the Oracle Access Governance Console. Every permission or role that needs to be assigned to a user must be processed through an approval workflow. As a resource administrator, you must design the workflow by specifying the required approval level and the number of approvers. Later, as a permissions manager, you can use these workflows to obtain approvals before assigning or revoking user privileges

For example, when a user requests access to an access bundle via a self-service system, the approval workflow associated with that request triggers a notification via email to the approvers, who then review the request and approve it, reject it, or can request for more

information. The result of the approval process is updated in the permissions management system.

> **Note:**
>
> The approver/manager can also log into the self-service system and review the pending approvals.

Navigate to Approval Workflow

1. Sign in to the Oracle Access Governance Console with a user assigned with *Service Administrator* role.
2. Click the icon and select **Access Controls**, and then **Approval Workflows**. You will be navigated to the *Approval Workflows* screen, where you can view and manage the existing workflows.
3. Click the **Create approval workflow** button.
   You will be navigated to the *Create a new approval workflow* screen, from which you can create and configure approval workflow.

Create Approval Workflow

1. In the *Create a new approval workflow* page, click the  icon.
   You will be navigated to the *Add a new approver* step.

> **Note:**
>
> You can add only one type of approver at a time.

Build Approvals

1. Select the type of approval from the **Which type of approval?** dropdown list.
   The available values are:
   - Benificiary
   - Beneficiary's Manager
   - Custom User
   - Identity Collection
   - Management Chain
   - Owner

> **Note:**
>
> The other options on the page change depending on the type of approval selected. We have selected the approval type as *Management Chain* for this workflow as an example.

2. Enter the following details in the *Add a new approval* pop-up window and then click **Add**.

Field	Description
Management Chain Ceiling	Select levels up in the management chain to be involved in the approval process.
Advanced settings	
How many hours between notifications?	Enter the number of hours within which a reminder must be sent to the approver.
How many hours to wait before escalating the approval request?	Enter the number of hours to wait prior to escalating an approval request.
Which identity collection should be excluded from escalations?	Select the identity collection that should be excluded from the escalations. **Note:** The group selected here will not receive any approval requests triggered by an escalation.
Should the approval request have an expiration time?	Add time after which the approval request will expire. Default is **No**.
Should the workflow continue when the approval result is rejection.	If **No** then, if any level of the workflow returns a rejection result, the workflow ends. If **Yes** then the workflow continues.

The entered details will be displayed on the *Create a new approval workflow* screen.

3. In the *Create a new approval workflow* screen, click:
   - **Add parallel**: To add a parallel approver. This adds an approver that will be requested at the same time as the currently selected approver.
   - **Add next**: To add a next level of approver. This adds an approver that will be requested after the current approver completes.

Add Parallel Approver

1. In the *Create a new approval workflow* screen, click the **Add parallel** button.
2. Enter the following details on the screen, and then click **Add** button.

Field	Description
Which type of approval?	Select the type of approval. **Note:** In this workflow, we have selected the type of approval as *Management Chain*.
Management Chain Ceiling	Select number of levels up in the management chain to be involved in the approval process.
How many levels up in the management chain should the approval go?	Select the level that should be involved in the approval process. **Note:** This field name changes based on the type of management chain ceiling selected.
Advanced settings	
How many hours between notifications?	Enter the number of hours within which a reminder must be sent to the approver.
How many hours to wait before escalating the approval request?	Enter the number of hours to wait prior to escalating an approval request.
Which identity collection should be excluded from escalations?	Select the identity collection that should be excluded from the escalations. **Note:** The group selected here will not receive any approval requests triggered by an escalation.
Should the approval request have an expiration time?	Add time after which the approval request will expire. Default is **No**.
Should the workflow continue when the approval result is rejection.	If **No** then, if any level of the workflow returns a rejection result, the workflow ends. If **Yes** then the workflow continues.

After adding a parallel approver, in the *Create a new approval workflow* screen, click:

- **All**: If you want all the approvers to approve the request
- **Any**: If you want any of the approvers to approve the request

Add Next Level of Approver

1. In the *Create a new approval workflow* screen, click the **Add next** button.
2. Enter the following details on the screen, and then click **Add** button.

Field	Description
Which type of approval?	Select the type of approval. **Note:** In this workflow, we have selected the type of approval as *Management Chain*.
Management Chain Ceiling	Select number of levels up in the management chain to be involved in the approval process.
How many levels up in the management chain should the approval go?	Select the level that should be involved in the approval process. **Note:** This field name changes based on the type of management chain ceiling selected.
Advanced settings	
How many hours between notifications?	Enter the number of hours within which a reminder must be sent to the approver.
How many hours to wait before escalating the approval request?	Enter the number of hours to wait prior to escalating an approval request.
Which identity group should be excluded from escalations?	Select the identity group that should be excluded from the escalations. **Note:** The group selected here will not receive any approval requests triggered by an escalation.
Should the approval request have an expiration time?	Add time after which the approval request will expire. Default is **No**.
Should the workflow continue when the approval result is rejection.	If **No** then, if any level of the workflow returns a rejection result, the workflow ends. If **Yes** then the workflow continues.

Add Details

With the *Add details* step, you can add a name to the approval workflow and give a short description of the workflow.

To add details:

1. Enter the name of the workflow in the **What do you want to call this approval workflow?** field.

2. Enter a short description of the workflow in the **How do you want to describe this approval workflow?** field.
3. On entering the details, click **Next** to go to the Review and submit step.
4. Optionally, you can click:
   - **Save Draft**: To save your changes and later come back and edit the workflow or details
   - **Cancel**: To cancel the current process.
   - **Back**: To go back to the previous step.

Add Primary and Additional Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

No special application roles are necessary for assigning resource ownership. Any Oracle Access Governance active user can be assigned as the owner of the resources. All the owners can read, update, or delete the resources that they own. However, the *Primary Owner* is assigned as the access reviewer when you choose the **Owner** template in the approval workflow for performing *Ownership reviews* in Campaigns. For more information, refer Types of Access Reviews Offered by Oracle Access Governance.

For assigning resource ownership, you must have active Oracle Access Governance users. When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

   You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Review and Submit

The Review and submit step displays the information you have added in the previous steps. In this screen, you can click:

- **Publish**: To publish the workflow
- **Save Draft**: To save your changes and later come back and edit the workflow or details
- **Cancel**: To cancel the current process
- **Back**: To go back to the previous step

Users with the *Access Control Administrator* or *Administrator* application role can monitor and manage the approvals using the Oracle Access Governance Console.

On creating a new approval, notification will be sent to the approvers via email, or you can either click on the link from the email or can log in to the self-service portal to access the existing approvals.

View Workflow Details

1. Login to the Oracle Access Governance Console with the *Access Control Administrator* role.
2. Navigate to the Approval Workflows screen.
   A list of existing approvals defined in the system displays.
3. Click ⋯ corresponding to the approval request.
4. From the pop-up menu click **View details**.
5. In the details screen, you can click on the ⋯ corresponding to each step to view further details.

Edit Workflow Details

1. In the Approval Workflow screen, click ⋯ corresponding to the approval request.
2. From the pop-up menu click **Edit**.
   Edit approval workflow screen appears.
3. In the **Build Approvals** step, click ⋯ corresponding to the required entity.
4. From the pop-up menu click **Edit**.
5. Modify the required details in the subsequent screens.
6. Optionally, you can click ⋯ corresponding to the **Single Approver** or **All approvers required** entity to either delete the approver, add next approver, or to add a parallel approver to the workflow.
7. Navigate to the **Add Details** step.
8. Modify the required details, and click **Next**.
9. Navigate to the **Review and Submit** step and click:
   - **Publish**: To publish the edits
   - **Back**: To do any corrections, if required
   - **Save Daft**: To save the workflow details
   - **Cancel**: To retain the existing workflow details

Delete Workflow

1. In the Approval Workflow screen, click ⋯ corresponding to the approval request.
2. From the pop-up menu click **Delete**.
3. In the confirmation pop-up, select:
   - **Confirm**: To remove the workflow or
   - **Cancel**: To retain the workflow

Disable Workflow

1. In the Approval Workflow screen, click ⋯ corresponding to the approval request.
2. From the pop-up menu click **Disable**.
3. In the confirmation pop-up, select:
   - **Confirm**: To disable the workflow or
   - **Cancel**: To retain the workflow

5
Integrate

Identity Orchestration is an Oracle Access Governance framework that brings together diverse Authoritative and Managed Systems by supporting low-code integrations. It facilitates data transformations and correlation rules which ensures data coherence, extracts the required identity data from various systems into Oracle Access Governance, and performs fulfillment through account provisioning.

The entire orchestration process involves:

- Integrating with various on-premises or cloud systems through low-code integration.
- Extracting or ingesting only the required information (identity attributes, permission assignments, and policies) into Oracle Access Governance.
- Transforming and correlating the ingested data, both identity and account attributes, to build a composite identity profile and account information.
- Processing the identity data and using it for access controls, access reviews, workflows, etc.
- Provisioning, and synchronizing data between the Orchestrated systems.

Significance of Identity Orchestration: Why Modern Identity Orchestration is Essential for your Enterprise

Identity Orchestration is crucial for a complex and dynamic IT ecosystem that may include the distributed nature of IT infrastructure, deployments across on-premises, demilitarized zone (DMZ), multi cloud, and IoT environments. Without this, enterprises face critical issues such as limited visibility of identity-related activities, fragmented access control, operational inefficiency, and a higher likelihood of security threats and compliance issues.

Managing identities and their respective accesses require seamless identity and access orchestration for effective identity lifecycle management, governance and compliance. Modern Identity Governance systems, such as Oracle Access Governance, offer a holistic Identity Orchestration system that provides low-code integrations, data correlation and data transformation capabilities along with fulfillment. This enables thorough access discovery, comprehensive insights into identity profiles and clearly stated access controls, access reviews, and micro-certifications.

Integrations in Oracle Access Governance

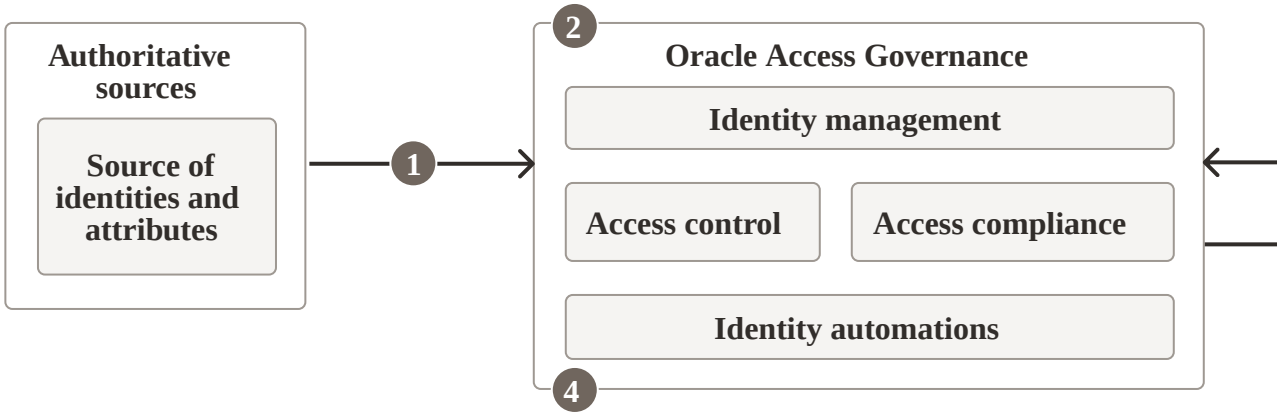
Oracle Access Governance simplifies identity orchestration by offering a wide range of specialized and generic out-of-the-box integrations, requiring minimal configurations.

- **Specialized Integrations**: Integrations for specific applications, providing application-specific use cases. For example, integration with Oracle Human Capital Management (HCM), Microsoft Entra ID, Microsoft Teams, etc.

- **Generic Integrations**: Integrations for constrained or sensitive applications, or for applications with unsupported data structures. You can achieve the integration by using a Flat File or Generic Rest API, offering flexibility and broader compatibility.

Oracle Access Governance carries out integrations either through API (direct integration) with cloud services and systems in public domains or using an agent, which is a downloadable docker image, for systems behind firewalls. These systems and applications can be integrated either as *Authoritative Sources* or *Managed Systems*.

Identity Orchestration Functional Overview



Let's understand the steps involved:

1. **Identity Data Synchronization + Correlation Rules + Inbound Data Transformation**: In the first step, there's synchronization of identity data from Authoritative Sources along with execution of correlation rules and inbound transformation on the ingested identity data. This is where Oracle Access Governance identities are created.
2. **Identity Profile + Identity Attributes**: In the second step, a composite identity profile is built by customizing and configuring identity attributes within Oracle Access Governance.
3. **Correlation Rules + Inbound Data Transformation + Account Reconciliation**: In the third step, there's execution of correlation rules and inbound transformation on the ingested account and permissions data from Managed Systems. During this process, your accounts are reconciled against identities. This is where Oracle Access Governance accounts are created, and are used for performing provisioning operations.
4. **Identity lifecycle + Access control + Access reviews**: In the forth step, you can perform usual Oracle Access Governance features, such as managing identity lifecycle, executing access reviews, setting up access controls and approval workflows within Oracle Access Governance.
5. **Outbound Data Transformation + Account Provisioning**: At last, **Oracle Access Governance** supports outbound data transformations which uses identity attributes to define account attributes for provisioning in the Managed Systems. For example, applying default values to null values or changing format of an attribute to maintain coherence throughout provisioning process.

Authoritative Source and Managed System

Based on the type of identity and access data extracted from systems or applications, Oracle Access Governance segregates the systems into:

- **Authoritative Source**: Trusted source of identity data and identity attributes that can be used by Oracle Access Governance to load and manage identity data. A few examples can be Oracle Identity Governance, Microsoft Entra ID (formerly known as Azure Active Directory), or any HR system to manage identity data and its attributes, such as email address, username, location, or department.
- **Managed System**: Applications and services containing accounts and respective access privileges but do not serve as a trusted source of identities in your enterprise information, for example, Oracle Database User management, Salesforce, and Microsoft Teams. By establishing an orchestrated system, Oracle Access Governance manages user accounts and access permissions for these applications leveraging the defined access controls (including access request, RBAC, ABAC, and PBAC).
- **Authoritative Source and Managed System**: Systems and applications can fulfill both roles, serving as the authoritative source for identity data while also acting as a Managed Systems for governing access.

Data Transformation and Correlation Rules

The key tenets of seamless identity orchestration include:

- **Correlation Rules**: You can leverage correlation or matching rules to match the identity data ingested from different Authoritative Sources, and thus build a composite identity profile. Similarly, during data ingestion from Managed Systems, multiple accounts may exist for an identity. You can match the account data with the respective identities to associate the user accounts ingested from Managed Systems with the identity. For example, you can match User Login coming from the Orchestrated System with Employee user name ingested in Oracle Access Governance.
- **Inbound Data Transformations**: Applications, whether Authoritative Sources or Managed Systems, may present data in different formats. During the data ingestion process from Authoritative Sources to Oracle Access Governance, you can transform the identity data to enhance the identity profile information using Inbound Transformation rules. For example, you may want to concatenate employee number with the first name to set a display name in Oracle Access Governance. Similarly, during the data ingestion from Managed Systems, you can define or customize account data using the Inbound Transformation rules. For example, during rebranding of a product, you may want to change the application display name to some other fixed value.
- **Outbound Data Transformations**: Oracle Access Governance offers Outbound Transformation rules, where you use identity attributes to define account attributes for account provisioning in the Managed Systems. For example, you can set organization having null value to some default value.

To summarize, Identity Orchestration is a vendor-agnostic solution of Oracle Access Governance for today's heterogeneous environments that works with all the leading Identity Providers (IDPs) or services to secure your IT infrastructure.

Identity Orchestration components serve as the building blocks that unify diverse systems and effectively manage the identity lifecycle for your enterprise. These include Oracle Access Governance agent for indirect integrations, inbound and outbound data transformation rules for ensuring data coherence, correlation rules to build composite profiles, and Orchestrated System Resources to effectively manage resources and control sources used to load the data.

Access Governance Agent

The Oracle Access Governance agent is a downloadable docker image, which allows Oracle Access Governance to synchronize continuously or periodically with Authoritative Sources or Managed Systems where a direct connection is not available.

The agent runs scheduled distributed extract-transform-load (ETL) jobs to perform full or incremental synchronization of remote identity data, such as users, roles, application instances, entitlements, and entitlement assignments, to Oracle Access Governance. Once registered and installed, the agent can be monitored via the Oracle Access Governance Console. The agent runs in a docker environment located in your local (customer) environment. This environment should meet the following prerequisites:

- Installation of Docker or Podman
- Allow connection to the customer's target identity database
- Allow connection to the customer's Oracle Access Governance instance hosted in Oracle Cloud. If required, this connection can be made through a web proxy.

The agent extracts data, picked up by the Oracle Access Governance ingestion service, and is loaded into Oracle Access Governance for consumption.

Fulfillment requests (provisioning of accounts or closed-loop access remediation) are passed on to Managed systems. For example, on completion of access review campaigns, any permissions that have been revoked in Oracle Access Governance will be remediated by raising a revoke operation in the Orchestrated system. This revoke request will be passed to the Managed system via the agent.

> **Note:**
>
> Agents are applicable only in cases where a direct connection cannot be established with Oracle Access Governance. Typically, you will need an agent when integrating with the on-premises systems. The Oracle Access Governance agent acts as an arbitrator supporting the synchronization of Authoritative Sources or Managed Systems and Oracle Access Governance.

Data Transformations

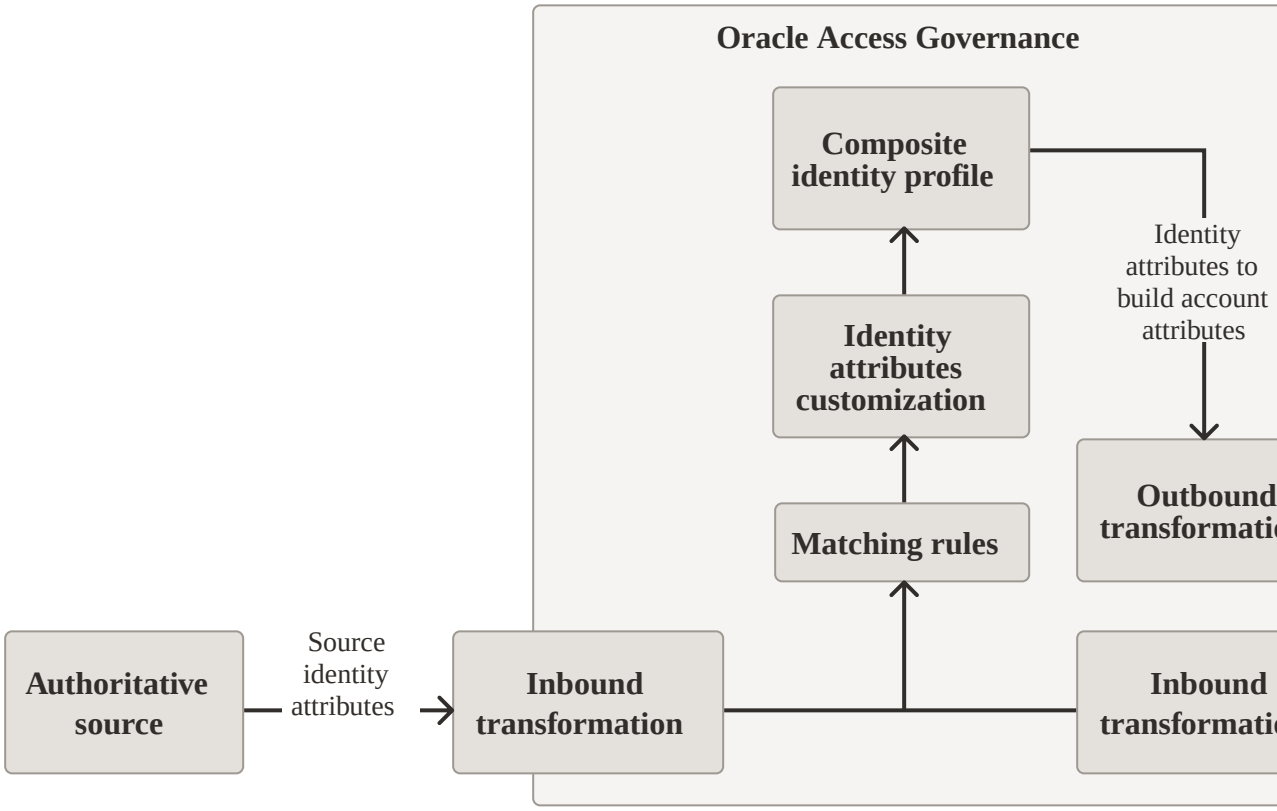
Different systems represent data differently. Oracle Access Governance allows you to manipulate and transform incoming identity and account data from Authoritative Source or Managed Systems, or outgoing data being provisioned to Managed Systems. You can modify data values to meet your requirements, such as including derived values or ensuring consistent formatting. This ensures data coherence and system unification.

You can apply data transformations to:

- Enhance incoming identity attributes being ingested from an Authoritative Source into Oracle Access Governance using inbound transformation rules. For example, you can set the username to uppercase, or concatenate the family name with the name to create a display name.
- Enhance incoming account attributes being ingested from Managed Systems into Oracle Access Governance using inbound transformation rules. For example, you can concatenate username with default domain to set the primary email within Oracle Access Governance.

- Customize composite identity attributes built up in Oracle Access Governance so that you can match incoming attributes. For example, Database User Management applications store username in a specific format and, as it contains only account data, you would need to customize the composite identity attributes in Oracle Access Governance so that incoming DBUM username matches with identity attributes available in Oracle Access Governance.
- Define account attributes by enhancing identity attributes for account provisioning by Oracle Access Governance into Managed Systems. For example, setting *jobDescription* to some fixed value, in case you have a null value.

Transformation Types and its Workflow



Let's see the types of transformations available in Oracle Access Governance and its workflow:

- You first integrate an Authoritative Source with Oracle Access Governance by adding an Orchestrated system. Here, you can execute inbound transformations on the source identity attributes data during the data ingestion process. For example, you may integrate Oracle HCM as an Authoritative Source and apply inbound transformations to create a display name from the full name. These transformations are unique and specific to each Orchestrated system.
- Once you integrate Authoritative source systems, internally, a composite identity profile is constructed that contains **Core** and **Custom** attributes within Oracle Access Governance. This composite identity profile can contain identity attributes from various Authoritative sources that you have integrated. For example, identity in Oracle Access Governance can contain attributes *jobCode* ingested from Oracle HCM and *department* ingested from Flat File. In the scenario where the same attribute is available in more than one Authoritative source, you have the option to select and change the Orchestrated system from which you want to bring this attribute value. Refer to Manage Identity Attributes for more details. This

composite identity profile acts as a source of truth for Oracle Access Governance to perform various governance and provisioning operations.

- Next, you integrate Managed Systems to load account attributes. These account attributes are matched against composite identity attributes available within Oracle Access Governance.
- For outbound provisioning, you manipulate the composite identity attributes available in Oracle Access Governance for accurate account provisioning into Managed Systems.

Inbound Data Transformation

Inbound data transformations allow you to control how attribute values are transformed when they are received from an orchestrated system into Oracle Access Governance. When you load data from an Orchestrated system, identity and account attributes will be imported into Oracle Access Governance. During the data load or data ingestion process, it is possible to apply data transformations to the attributes.

An example use case might include some of the following:

- Populate the primaryEmail identity attribute in Oracle Access Governance by concatenating FirstName, ., LastName, @mydomain.com. This is how you achieve this in Oracle Access Governance.

```
user.getFullName().getGivenName().concat('.',user.getFullName().getFamilyNa
me())+'mydomain.com'
```

- Set the value of an attribute to another value if the value coming from the orchestrated system is null. An example might be if the organization is null then set the value in Oracle Access Governance to a fixed value.

```
user.getOrganization() != null && user.getOrganization().getDisplayName() !
= null ?
      user.getOrganization().getDisplayName() : 'Oracle Access Governance'
```

Outbound Data Transformation

Identity Orchestration includes the ability to provision accounts using the Request Access (self service), Request Access (for others), or policy based access functionality. As part of this process, it is possible to apply data transformations to the data provisioned into the Managed system account. When provisioning operations such as *Change Password* or *Create Account* are triggered in Oracle Access Governance, data transformation rules can be invoked which derive values from the identity data and transform the data using strings and other manipulations, so that provisioning of accounts is done against correct identities into Managed system.

An example use case might include some of the following:

- Populate the **workEmail** attribute in the provisioned Managed system with the identity **primaryEmail** attribute value.
- Create a value for the **displayName** attribute in the provisioned Managed system by concatenating the identity **title**, **userName**, and **employeeNumber**.
- Set the value of an attribute to another value if the identity input value is null. An example might be if **organization** is null then set the value to a fixed value in the provisioned Managed system account.

Identity Attributes: Customizing Composite Oracle Access Governance Identity Profile by Applying Transformation Rules

When you load identity attributes from various Authoritative Sources, a composite identity profile is built in Oracle Access Governance, containing identity attributes from various sources. When you load account attributes from Managed Systems, the values available in this composite identity profile are matched with account attributes (Identity Account Matching). In special circumstances, you may have to apply additional transformation rules on this composite identity profile so that it can match the incoming data from various systems. Typically, you require this type of transformation to customize an identity attribute within Oracle Access Governance so that it matches the account attributes incoming from Managed Systems.

> **Note:**
>
> Inbound Transformations are specific to Orchestrated System and Identity Attribute Customization is applied on the composite identity profile built in Oracle Access Governance. For example, if you have applied inbound transformation on *JobCode* ingested from Oracle HCM but you have selected *Flat File* as the source for this attribute in the Manage Identities page, then this transformation won't have any impact on the value available in Oracle Access Governance.

Scenario

For example, your Database User Management (DBUM) may store an attribute, such as a username in a very specific format, different from the username available in the composite identity profile within Oracle Access Governance. To match a DBUM account with the identity that exists in Oracle Access Governance, you need to customize the Identity Attribute by applying transformation rules. As DBUM is a Managed System account, you cannot manipulate identity data by applying inbound transformation rules. That's where you need to customize composite Identity Attributes. For example, when you connect MySQL DBUM, Oracle Access Governance adds an internal identity attribute *userNameMysql*, with the applicable transformation rules. Then you can match username incoming from My SQL DBUM with *userNameMysql* identity attribute. The same rule is applied on the outbound transformation side so that the provisioning of accounts is done accurately.

Authoritative Source	Composite Identity Attribute	Managed System
• **Identity Attribute**: username • **Value**: *John.Doe@o.com*	• **Identity Attribute**: username • **Value**: *John.Doe@o.com*	• **Account Attribute**: userLogin • **Value**: *John_Doe@o.com*
	• **Internal Identity Attribute**: *userNameMysql* • **Value**: *John_Doe@o.com*	

The Identity attribute *username* has a different value than Account Attribute *userLogin* and we cannot match these attributes. The composite identity attributes transformation will transform *John.Doe@o.com* to *John_Doe@o.com* and store in the *userNameMysql* created within Oracle Access Governance. Once that is done, the incoming value, *John_Doe@o.com*, will match.

> **Note:**
>
> As a best practice, we recommend using inbound transformations to transform an identity attribute and limit the use of applying transformation rules on the composite identity profile. It may interfere with the provisioning of accounts.

Matching Rules

Oracle Access Governance leverages correlation, or matching, rules to match the identity data ingested from different Authoritative sources, and thus build a composite identity profile. Similarly, during data ingestion from Managed Systems, multiple accounts may exist for an identity. In this case, account data needs to be ingested and matched to the respective identities. Account matching rules can be leveraged to associate user accounts ingested from downstream applications with identities in Oracle Access Governance. In case, you have a system that acts both as Authoritative Source and Managed System within Oracle Access Governance , then you can implement identity matching and account matching for the same system.

You can easily build these correlation rules in Oracle Access Governance using identity and account attributes. If an account cannot be automatically matched to an identity, Oracle Access Governance creates a micro-certification for this unmatched account, so that it can be manually matched with the identity or remediated from the Managed System.

Types of Matching Rules

When data is received from an Orchestrated system, Oracle Access Governance, checks to see if the data matches data already onboarded as an identity or account. Oracle Access Governance supports the following matching types:

- **Identity matching**: This match checks if an incoming identity matches an existing identity or is new to Oracle Access Governance. If it is a match, then the incoming data is correlated with the existing identity. If there is no match, then the data is used to create a new identity in Oracle Access Governance.
- **Account matching**: This match checks if an incoming account matches with an existing identity. If there is a match the account information is correlated with the matching identity. If no match is found, then the account is flagged as unmatched.
- **Manually matched accounts**: If you have manually matched an unmatched account, then you will see those manually matched accounts under these.

Orchestrated System Resources

You can determine which resources you want to ingest from systems allowing full control of the source used to load data into Oracle Access Governance. You can manage which resources are populated from orchestrated systems. This functionality is very specific to Oracle Identity Governance.

A typical use case might be where you have identity data managed by Oracle Identity Governance (OIG), and you want governance in a hybrid fashion till the time you migrate completely to the cloud environment. By default, all resources ingested from the Authoritative Source and Managed System will be available to Oracle Access Governance. As you add direct connections between Oracle Access Governance and systems, you can remove these from your primary governance system to avoid duplication of data.

We have outlined eight essential tasks from adding an orchestrated system, transforming your data, implementing matching rules, to finally provisioning your accounts using Oracle Access Governance. These tasks serve as a checklist to ensure that integrations and operations between your system and Oracle Access Governance are streamlined and efficient.

Identity Orchestration Process Flow

To manage Identity Orchestration using the Oracle Access Governance Console, administrators need to perform the following tasks:

1. **Add an Orchestrated System**: You can integrate with a system of your choice by entering basic and configuration details. To do this, you first need to add an Orchestrated system in the Oracle Access Governance Console. Whenever you integrate an Authoritative Source or Managed System in the Oracle Access Governance Console, it is termed as an Orchestrated system in Oracle Access Governance. To know how to add orchestrated system within the Oracle Access Governance Console, see Add Orchestrated Systems.
2. **Validate the Connection**: Once you have added an Orchestrated system, you must test and verify when a new connection is established or when you update the connection settings. There are various activities involved once you have added an Orchestrated System. For additional details on activities, see View Activity Log.
3. **Configure Inbound Data Transformation Rules**: Different systems represent identity and access data in different data formats schema or formats or businesses may have specific data standards. To make data compatible between systems, you need to apply rules to transform the data coming into (data ingestion) or going out of (account provision) Oracle Access Governance. With data transformation, you can handle null values, aggregate, concatenate, normalize the data, etc. by writing JavaScript methods. To know how to apply data transformation in Oracle Access Governance, refer Configure Settings for an Orchestrated System and read more about data transformation and identity customization rules in the Data Rules to Customize and Transform Identity and Account Attributes article.
4. **Configure Correlation or Matching Rules**: To build a composite profile, you can match data incoming from different authoritative sources, by configuring matching or correlation rules. You can also match accounts with identities by configuring the account matching rules. Accounts that do not match any identities are tagged as Unmatched Accounts in Oracle Access Governance. Read more about Unmatched Accounts to understand how Oracle Access Governance handles unmatched or orphan accounts.
5. **Activate Identities**: Once the data load operation is successful, you must activate these identities within the service, and flag identities as either Workforce or Consumer users. Read more in the Manage Identities article.
6. **Configure Identity Attributes**: You can configure Identity attributes (Core or Custom) to perform various functions, such as running access review campaigns, choosing identities for identity collections, or applying attribute conditions to enable/disable the available identity data set. Read more about attributes in the View and Configure Identity Attributes article.

> **Note:**
>
> Post the tasks, you can use other Oracle Access Governance features, such as running campaigns for access reviews, managing permissions using the access control framework, viewing enterprise-wide insights, delegating tasks, setting up business processes by creating approval workflows, and so on.

7. **Configure Outbound Data Transformation Rules**: You can then configure outbound transformation rules using the identity attributes to define account attributes for provisioning (account provisioning) in Oracle Access Governance. To know how to apply data transformation in Oracle Access Governance, refer Configure Settings for an Orchestrated System
8. **Execute Provisioning of Accounts**: Finally, Oracle Access Governance performs the fulfillment process by provisioning accounts, or by sending out review decisions for closed-loop access remediation.

In some cases, an orchestrated system does not have a direct connection to Oracle Access Governance and requires an agent to enable data transfer between Oracle Access Governance and the orchestrated system.

Register and Download the Oracle Access Governance Agent

To enable an orchestrated system to connect to Oracle Access Governance, you need to enter integration details and credentials for the system and build an agent specific to your environment.

1. In a browser, navigate to the Oracle Access Governance service home page and log in as a user with the *Administrator* application role.
2. On the Oracle Access Governance service home page, click on the  icon and select **Service Administration** and then **Orchestrated Systems**.
3. Select the **Add a connected system** button, to navigate to the **Add n orchestrated system** page to begin the configuration.
4. On the **Select system** step, select the system you want to integrate with Oracle Access Governance, and then click **Next**.
5. On the **Enter Details** step, enter the basic details, such as name and description, for your orchestrated system, and then click **Next**.
6. On the **Configure** step, enter connection details for the orchestrated system:

> **Note:**
>
> The integration details will differ depending on the type of orchestrated system. For specific details related to each orchestrated system, refer to the Supported Integrations with Oracle Access Governance article.

7. Click **Next**. A message is displayed to download the agent. Select the **Download** link and download the agent zip file to the environment in which the agent will run.

8. Verify the package contents of your downloaded zip file.

   The contents of the agent package will look similar to the following:

   ```
   agent-package-<version>.zip
       - config.json
       - wallet
           - cwallet.sso
           - cwallet.sso.lck
      - agent-lcm
           - idm-agcs-agent-lcm
   ```

   The agent package does not contain the orchestrated system integration configuration. It connects to the Oracle Access Governance service instance and retrieves the information as required. This means that there is no need to download or reinstall the agent for subsequent integration configuration changes.

Prerequisites

Prerequisites for installation and running of an agent.

The following prerequisites should be met in order to install and run an agent.

1. The agent management script supports **docker** and **podman** as the container runtime. The agent management script auto-detects the container run time. If both are present, **podman** is selected.
2. The container runtime (docker/podman) should be configured to be run as a non-root user, the same as that which is used to install the agent.
3. Utilities:
   The agent requires the following operation system utilities:
   - **unzip**
   - **sed**
   - **awk**
   - **crontab**

   > **Note:**
   >
   > The agent installation user should have permission to use each of these utilities.

4. JDK: Agent requires **JDK 11.0.x**.
5. Enable processes for the OS user that starts the agent to 'linger' after the user's session is terminated. If this option is not enabled, when you terminate the user's session the agent process will stop and you will see errors when trying to communicate between Oracle Access Governance, the agent, and your orchestrated system.
   - To check if linger is enabled for your OS user check for a file with the same name as the user in the `/var/lib/systemd/linger` directory. If the file exists then this option is enabled:

     ```
     ls /var/lib/systemd/linger/oracle/<myuser>
     ```

- To enable linger for your OS user enable the systemd linger behavior:

```
loginctl enable-linger <myuser>
```

Sizing Virtual Machine/Host

The table below suggests values for sizing your orchestrated system agent VM or host for small, medium, and large scale implementations.

Parameter	Description	Small Scale	Medium Scale	Large Scale
CPU Cores	Number of CPU Cores.	2	4	8
Memory	Amount of memory (GB)	16	32	64

Install Oracle Access Governance Agent

A step-by-step process to install the Oracle Access Governance Agent with sample commands to run:

To install the downloaded agent into your local system, perform the following steps:

1. Unzip the downloaded agent to your local location.

   Contents of the unzipped agent should be:

```
agent-package-<version>.zip
  - config.json
  - wallet
      - cwallet.sso
      - cwallet.sso.lck
  - container-image
      - agent.tar.gz
```

2. Run the management script with the following parameters:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--agentpackage <PACKAGE_FULL_PATH> \
--install
```

   An example with default configuration would look like the following:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--agentpackage /access-governance/agent-management/agent-package-
<version>.zip \
--agentid myagent \
--install
```

An example with custom configuration would look like the following:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--agentpackage /access-governance/agent-management/agent-package-
<version>.zip \
--agentid myagent \
--config /access-governance/agent-management/config.properties \
--install
```

> **Tip:**
>
> If you encounter issues during copy-paste or an error while executing a script, such as an invalid format error, you may try to manually insert double hyphen ("-", ASCII value for hyphen is 45).

3. Start the agent with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--start
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--start
```

Verify Agent

Details how to verify the installation and operation of the orchestrated system agent.

To verify the installation of the orchestrated system agent, complete the following steps:

1. In the Oracle Access Governance Console, select the icon to display the navigation menu.
2. In the Oracle Access Governance Console, select **Service Administration** → **Orchestrated Systems** from the navigation menu.
3. On the **Orchestrated Systems** screen, the orchestrated system shows a status of **Waiting for initial integration**. Click on **Manage** → **Troubleshooting Checklist**.
4. The **Activity Log** at the bottom of the page will show the status of the Validate operation, **Pending** while the agent comes up. If the agent does not come up, check the agent install and operation logs for any issues.
5. Once the agent has come up, the status of the Validate operation will show as **Success**.

Agent Example Usage

Displays examples of usage of the agent management script.

Once you have successfully installed and verified your agent, you can start to manage the lifecycle. The `agentManagement.sh` script provides support for the start, stop, restart, uninstall, and upgrade operations.

Start the Agent

You start the agent with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--start
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--start
```

Stop the Agent

You stop the agent with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--stop
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--stop
```

Restart the Agent

You restart the agent with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--restart
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--restart
```

Uninstall the Agent

You uninstall the agent with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--uninstall
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--uninstall
```

Upgrade the Agent

You upgrade the agent with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--upgrade
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--upgrade
```

Enable Auto Upgrade

Enable auto upgrade with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh;  sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--enableautoupgrade
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh;  sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--enableautoupgrade
```

Disable Auto Upgrade

Perform this step only if absolutely necessary, as this can cause failures in the communication between the agent and the Oracle Access Governance service. If you perform this step and you see failures, immediately upgrade the agent by following the steps mentioned in the **Upgrade the Agent** example in Agent Example Usage.

Disable auto upgrade with the following command:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh;  sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--disableautoupgrade
```

For example:

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh;  sh agentManagement.sh \
--volume /access-governance/agent-management/volume \
--disableautoupgrade
```

Custom Jar Support

When integrating with Oracle Access Governance some orchestrated systems may need custom jar(s).For instance to communicate with DB2 and MYSQL database the respective driver jar is needed. Use the following steps to upload the custom jar.

1. Download the driver jar and place it in the `customJarsDirectory` path as specified in the `config.json`. For example, `config.json` may contain an entry, `customJarsDirectory : /app/data/customJars`, where `/app` is the agent volume.
2. Calculate the checksum of the downloaded driver jar using `SHA-512`.
3. On the Oracle Access Governance Console, go to **Service Administration** and then **Orchestrated Systems**.
4. On the **Connected Systems** page, select **Manage integration** for your orchestrated system.
5. Under **Configurations**, select the **Manage** button on the **Integration Settings** tile.
6. Update the orchestrated system configuration in the **Custom Jar Details** field. Provide the driver jar name and the checksum in the format `<jarName>::<checksum>`.
   For DB2 connected system, sample value in **Custom Jar Details**

   ```
   db2jcc.jar::c8520f145b428b1133b771bb2c70a6f0f546c9f0655f9de5de2e7b64d5ede786911ad50b5
   43846154fe373dead78d38fb6dded560e0de4c4e8ccbbf0a06b6c1e
   ```

7. Click **Save**.

Agent Management Operations

Lists details of the operations that the agent can perform and related parameter descriptions.

The orchestrated system agent can be managed using the `agentManagement.sh` script. This script can be downloaded from GitHub. The script supports `docker` and `podman`, it autodetects the container runtime available. If both are available, the script uses `podman`.

Operations

Operation	Description	Additional Information
`--install`	• Installs the downloaded agent package to the specified volume. • Loads the container image.	Use `--config` to use a custom configuration.
`--start`	• Starts the agent container. • Starts the agent daemon.	Use `--newcontainer` to start a new container. Use `--config` to use a custom configuration.
`--stop`	• Stops the agent daemon. • Stops the agent container.	
`--restart`	• Stops the agent daemon. • Stops the agent container. • Remove the agent container if `newcontainer` flag is set to **true**. • Starts the agent container. • Starts the agent daemon.	
`--uninstall`	• Stops the agent daemon. • Remove the agent container. • Clean up the volume.	
`--upgrade`	• Unzips new `agent-package.zip` in a temporary location. • Validates the package contents. • Loads the image from the new zip file. • Starts a temporary container using the new image and configuration. • If the temporary container has no issues then stop the container. • Stop the existing container. • Copy new configuration from the temporary location to the current location. This retains any customizations. • Starts the agent with the new image and configuration. • Starts the agent daemon.	The following changes require an upgrade. • Change in configuration `config.json` • Connector bundle change • Change in Wallet • Change of agent image The following changes will trigger a reconfigure operation which is handled by the agent framework. • Connector (same template version) • Connector (different template version) For more information, refer Upgrade an Agent.

Operation	Description	Additional Information
`--status`	Lists the following details of the agent: • Agent ID • Container runtime and version • Agent package • Agent version • Install location • Agent state	
`--enableautoupgrade`	Enables automatic upgrade by performing the following tasks: • Sets up a `cron` job to detect upgrades for any changes in target connectivity parameters, or in connector bundle code. • `cron` job runs every 30 minutes and upgrades the agent automatically if required.	
`--disableautoupgrade`	Disables automatic upgrades by removing the auto-upgrade `cron` job.	

Agent Parameters

Parameters

Parameter Name	Description	Mandatory	Default Value	Argument	Argument shorthand
__AGENT_ID__	Agent ID with which the agent container will run.	No	`agent-<hostname>-<timestamp>`	`--agentid`	`-ai`
Agent Package Location	Local Agent package location with the package name.	Yes		`--agentpackage`	`-ap`
Volume	Directory to persist agent data such as configuration, wallet, and logs.	Yes		`--volume`	`-pv`
New Container with start and restart	Create a new container. This parameter does not need a value..	No		`--newcontainer`	`-nc`

Parameter Name	Description	Mandatory	Default Value	Argument	Argument shorthand
Custom configuration	Provide custom configurations through a property file.	No		`--config`	`-c`
Auto Upgrade	Use this parameter with install operation to setup auto upgrade of the agent.	No	`true`	`--autoupgrade`	`-au`

Custom configuration is provided to the script via the `config.properties` file that has the following format:

```
idoConfig.httpClientConfiguration.connectionPool.maxPerRoute=15
idoConfig.httpClientConfiguration.connectionPool.maxTotal=15
idoConfig.httpClientConfiguration.connectTimeoutInSeconds=300
idoConfig.httpClientConfiguration.keepAliveTimeoutInSeconds=300
idoConfig.httpClientConfiguration.readResponseTimeoutInSeconds=300
idoConfig.httpClientConfiguration.proxyUri=
idoConfig.httpClientConfiguration.proxyUserName=
idoConfig.httpClientConfiguration.proxyUserPassword=
idoConfig.logLevel=info
idoConfig.maxJobRunningTimeInMinutes=180
idoConfig.numberOfPartition=3
idoConfig.numberOfOperationsPerPoll=5
idoConfig.numberOfOperationsWorkerThread=5
idoConfig.pollPauseTimeInMills=500
idoConfig.heartBeatIntervalInSeconds=30
idoConfig.sparkMaxResultSizeInGB=2
idoConfig.sparkExecutorMemoryInGB=2
```

Tuning Runtime Configuration

The table below lists the parameters for fine tuning the runtime configuration of the orchestrated system agent, and suggests specific values for small, medium, and large scale implementations.

Details of how to configure these parameters can be found in Agent Parameters.

Parameter	Description	Small Scale	Medium Scale	Large Scale
idoConfig.sparkMaxResultSizeInGB	Limit of total size of serialized results of all partitions for each action (e.g. collect) in bytes. Should be at least 1M, or 0 for unlimited. Jobs will be aborted if the total size is above this limit. Having a high limit may cause out-of-memory errors in driver (depends on spark.driver.memory and memory overhead of objects in JVM). Setting a proper limit can protect the driver from out-of-memory errors.	2	5	7
idoConfig.sparkExecutorMemoryInGB	Amount of additional memory to be allocated per executor process, in MiB unless otherwise specified. This is memory that accounts for things like VM overheads, interned strings, other native overheads, etc.	2	5	7
idoConfig.numberOfPartition	Number of partitions.	3	5	7

Troubleshooting Oracle Access Governance Agent

Learn how to address error messages and other problems you may see when configuring or using the Oracle Access Governance Agent.

Topics:

- Unexpected Agent Shutdown Due To Resource Constraints

Unexpected Agent Shutdown Due To Resource Constraints

If you start to hit resource limits on memory, CPU, or disk, the agent may unexpectedly shutdown. In order to bring the agent back up again cleanly you should restart the agent after rectifying the underlying issue.

Solution: Restart the agent using the `restart` command rectifying the underlying issue.

```
curl https://raw.githubusercontent.com/oracle/docker-images/main/
OracleIdentityGovernance/samples/scripts/agentManagement.sh -o
agentManagement.sh ; sh agentManagement.sh \
--volume <PERSISTENT_VOLUME_LOCATION> \
--newcontainer \
--restart
```

Using Orchestrated System of Oracle Access Governance, you can perform integrations with other systems by adding an orchestrated system, modify the connection settings, validate and load the data in Oracle Access Governance.

Add an Orchestrated System

You can integrate various systems, such as database, directory, applications, and cloud provider. The first step to integrate with the system of your choice is to add an Orchestrated system.

To add an Orchestrated system:

1. In the Oracle Access Governance Console, from the navigation menu, select **Service Administration → Orchestrated Systems**
2. Select **Add an orchestrated system** to add a new Orchestrated system, or select an existing orchestrated system from the list to manage existing orchestrated systems.

> **Note:**
>
> The integration details depend on the type of Orchestrated system. Refer to the Supported Integrations in Oracle Access Governance article for the appropriate documentation for the required Orchestrated system.

Manage an Orchestrated System

The **Orchestrated Systems** page of the Oracle Access Governance Console allows you to easily manage your orchestrated systems. You can view a list of the Orchestrated Systems configured in your service instance, including name, type, configuration mode, and status . You can also initiate a data load, update configuration settings, and enable/disable individual orchestrated systems.

To manage your orchestrated systems, navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. On the **Orchestrated Systems** page, you are able to view the orchestrated systems that you have configured in your Oracle Access Governance instance. The list of orchestrated systems displays the following details:
   - **Name**: The name of the orchestrated system.

- **Type**: The type of orchestrated system, for example, *Flat File*, *Database User Management (Oracle)*, or *Microsoft Active Directory*. A complete list of orchestrated system types supported by Oracle Access Governance can be found in Supported Systems.
- **Configuration mode**: The configuration mode that the orchestrated system is setup with.
- **Last updated**: Last date the orchestrated system configuration was updated.
- **Upcoming data load**: Date and time of any upcoming scheduled data load.
- **Status**: Current status of the orchestrated system. Values include:
  - Active
  - Disabled
  - Waiting for initial connection

Once you have reviewed the orchestrated systems in your Oracle Access Governance instance, you can select a specific orchestrated system and drill down into further information or update various configuration elements.

Select one of the following to view configuration of a specific orchestrated system:

- The orchestrated system link in the **Name** column.

- **Manage connection** from the navigation menu.

This displays the configuration page for the selected orchestrated system.

Manage Orchestrated System Resources

You can determine which resources are ingested from governance orchestrated systems, allowing full control of the source used to load data into Oracle Access Governance. This functionality is specific to Oracle Identity Governance.

To manage resources:

1. In the Oracle Access Governance console, access the navigation menu by selecting the icon. Select **Service Administration** → **Orchestrated Systems**.

2. In the **Orchestrated Systems** page, click on the icon for the governance orchestrated system you want to update, and select **Manage resources** from the drop-down list.
3. On the **Resources** page, you can see a list of *Connected resources* and *Disconnected resources*.
4. To disable a connected resource:
   a. Select the **Disconnect** icon, , for the resource you want to disable.
   b. A confirmation dialog displays, asking you if you are sure you want to disconnect the resource from the governance orchestrated system.

> **Note:**
>
> All information related to the resource will be removed, and you cannot reconnect the resource once it is disconnected.

c. If you want to remove the resource, click **Disconnect**. If not, select **No, keep connected**.

d. If the resource is disconnected, it will now display in the *Disconnected resources* section.

Initiate Data Load

You can initiate data load from orchestrated systems on-demand, using the Oracle Access Governance Console.

To initiate a data load from an orchestrated system, perform the following tasks.

1. From the Oracle Access Governance navigation menu icon , select **Service Administration → Orchestrated Systems**.
2. Select one of the following to view the configuration of a specific orchestrated system:
   - The orchestrated system link in the **Name** column.

     •••
   - **Manage integration** from the navigation menu.

   This displays the configuration page for the selected orchestrated system.
3. Select the **Load data now** button which will initiate a data load. You can track the status in the **Activity Log**.

View Activity Log

The activity log allows you to monitor the status of your orchestrated system.

To view the activity log:

1. From the Oracle Access Governance navigation menu icon , select **Service Administration → Orchestrated Systems**.

   •••
2. Select **View activity log**from the navigation menu to view the configuration of a specific orchestrated system. This displays the Activity log page for the selected orchestrated system, which which displays log details for the selected orchestrated system.

The activity log includes information on activities including the following:

- **Data load**: Initiates when the data is either run on-demand by the Administrator, or when data is auto-synced as per the system settings. Currently, the data automatically refreshes after 24 hours from the previous data load activity.

- **Full data load**: Initiates when the data is synced for the first time after the new integration is established.
- **Validate**: Initiates when a new integration is established or when you update the connection settings.
- **Revoke**: Initiates when an access reviewer revokes one or more user privileges in the access review tasks. This activity occurs to support closed-loop access remediation.
- **Schema discovery**: Initiates when a new integration is established, or when you select the Fetch attributes button in the *Identity Attributes* page.
- **Provisioning**: Create Account, Update Account, Add Child Data, Remove Child Data.

Disable an Orchestrated System

You can disable an orchestrated system to cease operations between Oracle Access Governance and the orchestrated system.

To disable an orchestrated system, perform the following tasks.

1. From the Oracle Access Governance navigation menu icon, select **Service Administration** → **Orchestrated Systems**.
2. Select one of the following to view the configuration of a specific orchestrated system:
   - The orchestrated system link in the **Name** column.
   - **Manage connection** from the navigation menu.

   This displays the configuration page for the selected orchestrated system.
3. Select the **Disable** button which will disable the selected orchestrated system.

With Oracle Access Governance, you can configure an orchestrated system by editing the integration settings, configuring notification settings, transforming inbound and outbound data for identity and account attributes, and applying matching or correlation rules to ensure integrated components work seamlessly together.

Modify Integration Settings for an Orchestrated System

You can configure the integration settings for your orchestrated system, using the Oracle Access Governance Console.

To update the integration details used by Oracle Access Governance to connect to an orchestrated system, perform the following tasks.

1. From the Oracle Access Governance navigation menu icon, select **Service Administration** → **Orchestrated Systems**.
2. Select one of the following to view the configuration of a specific orchestrated system:
   - The orchestrated system link in the **Name** column.

- **Manage connection** from the navigation menu.

This displays the configuration page for the selected orchestrated system.

3. From the **Configurations** section of the page, select **Manage** on the **Integration settings** tile. This will display the Integration settings page for your orchestrated system. The integration settings displayed are dependent on the type of orchestrated system you are updating.
4. Update the integration settings as required, and click **Save**.

Modify Account Settings for an Orchestrated System

You can configure the account settings for your orchestrated system to send notifications either to User or User manager whenever a new account is created. You can also choose to either disable or delete the account whenever an identity move or leaves your enterprise.

To update the account details used by Oracle Access Governance to connect to an orchestrated system, perform the following tasks.

1. From the Oracle Access Governance navigation menu icon , select **Service Administration → Orchestrated Systems**.
2. Select one of the following to view the configuration of a specific orchestrated system:
   - The orchestrated system link in the **Name** column.

   - **Manage integration** from the navigation menu.

   This displays the configuration page for the selected orchestrated system.
3. From the **Configurations** section of the page, select **Manage** on the **Account settings** tile. This will display the Account settings page for your orchestrated system.
   Enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

   Select where to send notification emails when an account is created. The default setting is **User**. You must select at least one of these options.

   - User
   - User manager

   When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:

   - Disable
   - Delete

   When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:

   - **Disable**: This sends an **Update Account** provisioning request to disable the accounts

- **Delete**: This sends a **Revoke** account provisioning request to delete the accounts

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, User Manager) when creating the orchestrated system. To update account settings for movers and leavers, you first need to create the orchestrated system, and then update the account settings.

If your orchestrated system is managing permissions then you can enable or disable Segregation of Duty violation checks. To do this select or unset the **Enable Risk Management and Compliance (RMC) integration for separation of duties check** checkbox.

4. Update the account settings as required, and click **Save**.

Configure Data Load Schedule Settings for Orchestrated Systems

Set how often data should be loaded and updated in Oracle Access Governance from the orchestrated system. Schedule timing and frequency by choosing specific days, hours, or minutes.

You can configure the timing and frequency for all orchestrated system except generic integration of Flat File and Oracle Cloud Infrastructure.

To configure data load settings:

Navigate to Data Load Settings

1. From the Oracle Access Governance navigation menu icon, select **Service Administration**, and then **Orchestrated Systems**.

2. For an orchestrated system, click the **More Actions** icon, and then select **Manage Integrations**.

Select Frequency

3. In the **Run every** field, choose a number to specify how often the data load should occur
4. In the **Frequency** drop-down, select one:
   - **Hours**
   - **Minutes**
   - **Days**

Limits have been applied to ensure reliable data is available and prevent outdated data. For example, the frequency cannot be less than 5 minutes.

Select Start Date

5. In the **Starting on** field, select the  date time icon to specify when the data load should begin, and then click **Done**.
6. Click **Save**. To save your settings for the orchestrated system, in the conformation pop-up box, click **Confirm**.

If the previous data load takes longer to complete, the next schedule load will be skipped. To avoid skipped syncs, ensure your settings allow enough time for each load to finish before the next one starts.

Add Primary and Additional Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

No special application roles are necessary for assigning resource ownership. Any Oracle Access Governance active user can be assigned as the owner of the resources. All the owners can read, update, or delete the resources that they own. However, the *Primary Owner* is assigned as the access reviewer when you choose the **Owner** template in the approval workflow for performing *Ownership reviews* in Campaigns. For more information, refer Types of Access Reviews Offered by Oracle Access Governance.

For assigning resource ownership, you must have active Oracle Access Governance users. When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

   You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Configure Identities or Email for Sending Orchestrated System Related Notifications

If an issue occurs in an orchestrated system during dataload, you want to be notified in good time so that you can investigate and resolve the issue. You can configure identities or an external email, to route notifications regarding your orchestrated system to assist with this.

To send orchestrated system-related notifications to your preferred identities or an external email address, you can configure Oracle Access Governance as required:

1. From the Oracle Access Governance service home page click on the  icon, and select **Service Administration** → **Orchestrated Systems**.
2. Select the orchestrated system you want to configure notifications for.
3. From the tiles in the **Configuration** section of the page, select **Manage** on the **Notification settings** tile.

4. In the **Which identities?** field, use the drop-down list to select identities in your Oracle Access Governance instance to send orchestrated system-related notifications to. You can have multiple identities as required.
5. In the **Email** field, add an email for any person external to your Oracle Access Governance instance (who does not have an identity in your system) who you would like to receive notifications. You can only add one external email address for orchestrated system-related notifications.

Match Identity and Account Attributes using Correlation Rules

Oracle Access Governance leverages correlation or matching rules to match the identity and account data and build a composite identity profile. To configure matching rules in Oracle Access Governance perform the following steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Select one of the following to view the configuration of a specific orchestrated system:
   - The connected system link in the **Name** column.

   - **Manage connection** from the navigation menu.

   This displays the configuration page for the selected orchestrated system.
3. From the **Configurations** section of the page, select **Manage** on the **Matching rules** tile. This will display the Matching rules page for your orchestrated system.
4. The tabs displayed depend on the configuration mode you selected when creating the orchestrated system, and by whether any unmatched accounts have been manually matched for this integration.
   - If you selected *This is the authoritative source for my identities.* then the **Identity matching** tab is displayed to set the matching rule for incoming identities.
   - If you selected *I want to manage permissions for this system.* then the **Account matching** tab is displayed to set the matching rule for incoming accounts.
   - If you selected both *This is the authoritative source for my identities.* and *I want to manage permissions for this system.* then both tabs are displayed.
   - If the orchestrated system selected has accounts which were unmatched, but have been manually matched, then the **Manually matched accounts** tab is displayed. You can unlink an account with the associated identity by selecting the

     Disconnect icon, or you can update the manual match by selecting the  **Edit** icon.
5. Select the tab you require to update identity matching rules or account matching rules.
6. Select one of the following conditions:
   - **All**: All rules must be matched in this case so order of the rules is not significant.

- **Any**: Any rule can, when met, produce a match. In this case order is significant as the matching rule will exit when a match is found. If you need to move a rule up the list you can select the menu for the rule, and select **Move up**.

7. Add a rule by selecting an **Equals** or **Not equals** operator.
8. Update the matching rules as required, and click **Save**.

Apply Inbound Transformations for Identity and Account Attributes

To modify the incoming data ingested into Oracle Access Governance, you need to apply inbound data transformations. To do so, perform the following tasks:

1. In the Oracle Access Governance Console, access the navigation menu by selecting the  icon. Select **Service Administration** → **Orchestrated Systems** .
2. Select the orchestrated system from the list which you want to configure inbound data transformation rules for.
3. Expand the **Configurations** drop-down menu and select the **Manage** button on the **Inbound data transformations** tile. The **Inbound data transformations** page displays a list of any rules that you have configured, and an option to add new attribute rules.
4. To create an attribute rule for your orchestrated system, select the **Add attribute rule** button.
5. In the **Add attribute rule** panel enter the following information to configure your rule.
   - **Which configuration mode?**: Select one configuration mode, from the drop down list, that you want this attribute rule to apply to.
     - **Authoritative source**: Authoritative Sources that contain identity data and its attributes.
     - **Managing permissions**: Managed Systems containing account information and permissions.
   - **Which attribute?**: Select the Oracle Access Governance attribute you want to apply the transformation to from the drop down list. The list of attributes available will depend on the orchestrated system type, and configuration mode you choose.
   - **Rule**: Enter the rule you want to apply to this operation/attribute.
   - Click the **Validate** button to check your rule. If the rule is valid then you will see a confirmation message and the rule will be marked as validated. If there is an issue with the rule, then you will see an error message and the rule will be marked as invalid. You cannot save your rule if it is marked as invalid.
   - When your rule is valid click **Add** to save your configuration.

Apply Outbound Transformations for Identity Attributes

To modify the outgoing data provisioned in Oracle Access Governance, you need to apply outbound data transformations. To do so, perform the following tasks:

1. In the Oracle Access Governance Console, access the navigation menu by selecting the  icon. Select **Service Administration** → **Orchestrated Systems**.

2. Select the orchestrated system from the list for which you want to configure the outbound data transformation rules.
3. Expand the **Configurations** drop-down menu and select the **Manage** button on the **Outbound data transformations** tile. The **Outbound data transformations** page displays a list of any rules that you have configured, and an option to create attribute rules.
4. To create an attribute rule for your orchestrated system, select the **Add attribute rule** button.
5. In the **Add attribute rule** panel enter the following information to configure your rule.
   - **Which operations**: Select one or more of the operations from the drop down list that you want this attribute rule to apply to.
     - Create Account
     - Change Password
   - **Which attribute?**: Select the attribute in the orchestrated system you want to apply the transformation to from the drop down list. The list of attributes available will depend on the orchestrated system type.
   - **Rule**: Enter the rule you want to apply to this operation/attribute.
   - Click the **Validate** button to check your rule. If the rule is valid then you will see a confirmation pop-up message and the rule will be marked as validated. If there is an issue with the rule, then you will see an error pop-up message and the rule will be marked as invalid. You cannot save your rule if it is marked as invalid.
   - When your rule is valid click **Add** to save your configuration.

Oracle Access Governance offers a various specialized and generic integrations. Specialized integrations involve integrations for specific applications, providing application-specific use cases whereas Generic integrations are commonly used when you want to integrate constrained or sensitive applications, or for applications with unsupported data structures. The connections can be direct, which is API-based, or indirect which are Agent-based.

You would need Agent-based connections where a direct connection cannot be established with Oracle Access Governance. Typically, you will need an agent when integrating with the on-premises target systems.

Supported Systems

Oracle Access Governance supports integrations with various types, such as databases, directories, Oracle and Non-Oracle applications.

Here's a list of systems that you can integrate with Oracle Access Governance:

Table 5-1 Orchestrated System Types/Systems

Type	System	Connection Type	Configuration Mode	Reference Documentation
Identity Governance System				

Table 5-1 (Cont.) Orchestrated System Types/Systems

Type	System	Connection Type	Configuration Mode	Reference Documentation
	Oracle Identity Governance	Agent-based	Authoritative Source	Integrate with Oracle Identity Governance
Cloud Service Provider				
	Oracle Cloud Infrastructure	API-based	Authoritative Source	Integrate with Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM)
Directory				
	Oracle Internet Directory	Agent-based	• Authoritative Source • Managed Systems	Integrate with Oracle Internet Directory
	Oracle Unified Directory	Agent-based	• Authoritative Source • Managed Systems	Integrate with Oracle Unified Directory
	Microsoft Active Directory	Agent-based	• Authoritative Source • Managed Systems	Integrate with Active Directory
	Microsoft Entra ID (formerly Azure Active Directory)	API-based	• Authoritative Source • Managed Systems	Integrate with Microsoft Entra ID (formerly Azure Active Directory)
Applications				
	Oracle Enterprise Business Suite (EBS) HRMS	Agent-based	• Authoritative Source • Managed Systems	Integrate with Oracle e-Business Employee Reconciliation (HRMS)
	Oracle E-Business Suite User Management	Agent-based	• Authoritative Source • Managed Systems	Integrate with Oracle e-Business User Management (UM)
	Oracle Fusion Cloud Applications	API-based	• Authoritative Source • Managed Systems	Integrate with Oracle Fusion Cloud Applications
	Oracle NetSuite	API-based	Managed Systems	Integrate with Oracle NetSuite
	Eloqua	API-based	Managed Systems	Integrate with Eloqua
	Primavera	API-based	Managed Systems	Integrate with Oracle Primavera

Table 5-1 (Cont.) Orchestrated System Types/Systems

Type	System	Connection Type	Configuration Mode	Reference Documentation
	Oracle Siebel CRM	API-based	• Authoritative Source • Managed Systems	Integrate with Oracle Siebel CRM
	Oracle Health EHR (formerly Cerner Millenium)	Agent-based	Managed Systems	Integrate with Oracle Health EHR (formerly Cerner Millennium)
	PeopleSoft	API-based	• Authoritative Source • Managed Systems	Integrate with Peoplesoft
		API-based	Managed Systems	Integrate with SAP Ariba
		API-based	Managed Systems	Integrate with SAP S4Hana
		API-based	• Authoritative Source • Managed Systems	Integrate with SAP SuccessFactors
Database Management System				
	Oracle Database	Agent-based	Managed Systems	Integrate with Database User Management (Oracle)
	Microsoft SQL Server	Agent-based	Managed Systems	Integrate with Database User Management (MSSQL)
	MySQL	Agent-based	Managed Systems	Integrate with Database User Management (MySQL)
	DB2	Agent-based	Managed Systems	Integrate with Database User Management (DB2)
	Database Application Tables (Oracle)	Agent-based	• Authoritative Source • Managed Systems	Integrate with Database Application Tables
	Database Application Tables (MSSQL)	Agent-based	• Authoritative Source • Managed Systems	Integrate with Database Application Tables
	Database Application Tables (MySQL)	Agent-based	• Authoritative Source • Managed Systems	Integrate with Database Application Tables

Table 5-1 (Cont.) Orchestrated System Types/Systems

Type	System	Connection Type	Configuration Mode	Reference Documentation
Other				
	Flat File	API-based	• Authoritative Source • Managed Systems	Integrate with Flat File
	Microsoft Teams	API-based	Managed Systems	Integrate with Microsoft Teams
	Generic REST	API-based	• Authoritative Source • Managed Systems	Integrate with Generic REST
		API-based	Managed Systems	Integrate with Arcon PAM

You can add rules to customize or transform identity and account attributes. These rules are written in JavaScript.

Objects

Attribute values for an outbound data transformation can be derived from the following objects:

Table 5-2 Objects (Outbound)

Object Name	Purpose	Example
requestAttributes	Provisioning request Attribute Object. These attributes are available when provisioning is via policy based, access bundle request, role based, or direct methods.	`requestAttributes.get('name')`
user	User Object. Use required getter on this to access any member.	`user.getName().getGivenName(), user.getUserName()`
application	Resource Object. Use required getter on this to access any member.	`application.getDisplayName()`

Attribute value for an inbound data transformation can be derived from the following objects:

Table 5-3 Objects (Inbound)

Object Name	Purpose	Example
user	User Object. Use required getter on this to access any member for Source of Identity.	`user.getName().getGivenName(), user.getUserName()`

Table 5-3 (Cont.) Objects (Inbound)

Object Name	Purpose	Example
account	Account Object. Use required getter on this to access any member for Manage Permissions.	`account.getDisplayName()`

Best Practices to Transform or Customize Identity and Account Attributes

Here are a few best practices and recommendations to consider:

- You can transform or customize identity and account attributes for Inbound data ingested from Authoritative Sources or Managed Systems. However, you can only transform identity (user) attributes for Outbound Data.
- Always perform a NULL check in rules for extracted values before using them otherwise it can lead to ingestion cycle failures on NULL references. This has to be done for both, *user* attributes object in Authoritative Sources and *account* attributes object in Managed Systems for inbound transformations.
- You cannot directly transform or assign value to attributes having array object data type, i.e. attributes returning a list of values, such as emails, photos, and addresses, but you can use them to modify/manipulate other user or account attributes. For example, to set the country value as the default value for the attribute location if a location is null, use:

```
user.getLocation() !=null  ? user.getLocation() : user.getAddresses()
[0].getCountry()
```

Authoritative Source Identity Object Attributes for Outbound Transformation

You can modify or alter the outbound data by applying data transformation rules to the data available or provisioned into the Orchestrated system. Here's a list of identity (user) attributes available for use in outbound data transformations.

Syntax to Fetch Identity Attributes for Outbound Data

These details can be fetched using the syntax:

```
get<FieldName>()
```

Example 5-1 Retrieve the user's given name

```
user.getName().getGivenName()
```

Table 5-4 Identity (User) Object Attributes for Outbound Data

Attribute	Sub Attribute	Data Type	Syntax
name		Reference	`user.getName()`
	formatted	String	`user.getName().getFormatted()`
	familyName	String	`user.getName().getFamilyName()`

Table 5-4 (Cont.) Identity (User) Object Attributes for Outbound Data

Attribute	Sub Attribute	Data Type	Syntax
	givenName	String	`user.getName().getGivenName()`
	middleName	String	`user.getName().getMiddleName()`
	honorificPrefix	String	`user.getName().getHonorificPrefix()`
	honorificSuffix	String	`user.getName().getHonorificSuffix()`
userName		String	`user.getUserName()`
displayName		String	`user.getDisplayName()`
description		String	`user.getDescription()`
primaryEmail		String	`user.getPrimaryEmail()`
userType		String	`user.getUserType()`
title		String	`user.getTitle()`
employeeNumber		String	`user.getEmployeeNumber()`
organization		Reference	`user.getOrganization()`
	value	String	`user.getOrganization().getValue()`
	ref	String	`user.getOrganization().getRef()`
	displayName	String	`user.getOrganization().getDisplayName()`
	resourceType	String	`user.getOrganization().getResourceType()`
department		String	`user.getDepartment()`
manager		Reference	`user.getManager()`
	value	String	`user.getManager().getValue()`
	ref	String	`user.getManager().getRef()`
	displayName	String	`user.getManager().getDisplayName()`
	resourceType	String	`user.getManager().getResourceType()`
status		String	`user.getStatus()`
jobCode		String	`user.getJobCode()`
state		String	`user.getState()`
risk		String	`user.getRisk()`
location		String	`user.getLocation()`
emails		List of Email	`emails = user.getEmails()` `email = user.getEmails() != null ? user.getEmails().get(0) : null`
	pendingVerificationData	String	`email.getPendingVerificationData()`
	primary	Boolean	`email.getPrimary()`
	secondary	Boolean	`email.getSecondary()`
	type	String	`email.getType()`
	value	String	`email.getValue()`

Table 5-4 (Cont.) Identity (User) Object Attributes for Outbound Data

Attribute	Sub Attribute	Data Type	Syntax
	verified	Boolean	`email.getVerified()`

addresses		List of Address	`addresses = user.getAddresses();`
			`address = user.getAddresses() != null?`
			`user.getAddresses().get(0) : null`

	country	String	`address.getCountry()`
	formatted	String	`address.getFormatted()`
	locality	String	`address.getLocality()`
	postalCode	String	`address.getPostalCode()`
	primary	Boolean	`address.isPrimary()`
	region	String	`address.getRegion()`
	streetAddress	String	`address.getStreetAddress()`
	type	String	`address.getType()`

phoneNumbers		List of PhoneNumber	`phoneNumbers = user.getPhoneNumbers();`
			`phoneNumber = user.getPhoneNumbers() != null?`
			`user.getPhoneNumbers().get(0) : null;`

	display	String	`phoneNumber.getDisplay()`
	primary	Boolean	`phoneNumber.isPrimary()`
	type	String	`phoneNumber.getType()`
	value	String	`phoneNumber.getValue()`
		Boolean	`phoneNumber.isVerified()`

Authoritative Source Identity Object Attributes for Inbound Transformation and Identity Attributes Customization

You can modify or alter the incoming data by applying data transformation rules during the data ingestion phase into the Orchestrated system. You can use the same set of attributes to customize composite identity profile constructed in Oracle Access Governance by transforming identity attributes.

Syntax to Fetch Identity Attributes for Inbound Data

The attribute details can be fetched using the syntax:

```
get<FieldName>()
```

Example 5-2 Retrieve the user's given name

```
user.getName().getGivenName()
```

Table 5-5 Authoritative Source Identity Attributes for Inbound Data

Attribute	Sub Attribute	Data Type	Syntax
fullName (for OIG/ICF)		Reference	`user.getFullName()`
	formatted	String	`user.getFullName().getFormatted()`
	familyName	String	`user.getFullName().getFamilyName()`
	givenName	String	`user.getFullName().getGivenName()`
	middleName	String	`user.getFullName().getMiddleName()`
	honorificPrefix	String	`user.getFullName().getHonorificPrefix()`
	honorificSuffix	String	`user.getFullName().getHonorificSuffix()`
name (for OCI)		Reference	`user.getName()`
	formatted	String	`user.getName().getFormatted()`
	familyName	String	`user.getName().getFamilyName()`
	givenName	String	`user.getName().getGivenName()`
	middleName	String	`user.getName().getMiddleName()`
	honorificPrefix	String	`user.getName().getHonorificPrefix()`
	honorificSuffix	String	`user.getName().getHonorificSuffix()`
userName		String	`user.getUserName()`
displayName		String	`user.getDisplayName()`
description		String	`user.getDescription()`
primaryEmail		String	`user.getPrimaryEmail()`
userType		String	`user.getUserType()`
title		String	`user.getTitle()`
employeeNumber		String	`user.getEmployeeNumber()`
organization		Reference	`user.getOrganization()`

Table 5-5 (Cont.) Authoritative Source Identity Attributes for Inbound Data

Attribute	Sub Attribute	Data Type	Syntax
	value	String	`user.getOrganization().getValue()`
	ref	String	`user.getOrganization().getRef()`
	displayName	String	`user.getOrganization().getDisplayName()`
	resourceType	String	`user.getOrganization().getResourceType()`
department		String	`user.getDepartment()`
manager		Reference	`user.getManager()`
	value	String	`user.getManager().getValue()`
	ref	String	`user.getManager().getRef()`
	displayName	String	`user.getManager().getDisplayName()`
	resourceType	String	`user.getManager().getResourceType()`
status		String	`user.getStatus()`
jobCode		String	`user.getJobCode()`
state		String	`user.getState()`
risk		String	`user.getRisk()`
location		String	`user.getLocation()`
compartmentId		String	`user.getCompartmentId()`
domainId		String	`user.getDomainId()`
domainOCID		String	`user.getDomainOCID()`
region		String	`user.getRegion()`
emails		List of Email	`emails = user.getEmails()`
	pendingVerificationData	String	`user.getEmails()[0].getPendingVerificationData()`
	primary	Boolean	`user.getEmails()[0].getPrimary()`
	secondary	Boolean	`user.getEmails()[0].getSecondary()`
	type	String	`user.getEmails()[0].getType()`

Table 5-5 (Cont.) Authoritative Source Identity Attributes for Inbound Data

Attribute	Sub Attribute	Data Type	Syntax
	value	String	`user.getEmails()[0].getValue()`
	verified	Boolean	`user.getEmails()[0].getVerified()`
addresses		List of Address	`addresses = user.getAddresses();`
	country	String	`user.getAddresses()[0].getCountry()`
	formatted	String	`user.getAddresses()[0].getFormatted()`
	locality	String	`user.getAddresses()[0].getLocality()`
	postalCode	String	`user.getAddresses()[0].getPostalCode()`
	primary	Boolean	`user.getAddresses()[0].isPrimary()`
	region	String	`user.getAddresses()[0].getRegion()`
	streetAddress	String	`user.getAddresses()[0].getStreetAddress()`
	type	String	`user.getAddresses()[0].getType()`
phoneNumbers		List of PhoneNumber	`phoneNumbers = user.getPhoneNumbers();`
	display	String	`user.getPhoneNumbers()[0].getDisplay()`
	primary	Boolean	`user.getPhoneNumbers()[0].isPrimary()`
	type	String	`user.getPhoneNumbers()[0].getType()`
	value	String	`user.getPhoneNumbers()[0].getValue()`
		Boolean	`user.getPhoneNumbers()[0].isVerified()`
photos		List of photos	`photos = user.getPhotos();`

Table 5-5 (Cont.) Authoritative Source Identity Attributes for Inbound Data

Attribute	Sub Attribute	Data Type	Syntax
	display	String	`user.getPhotos()[0].getDisplay()`
	primary	Boolean	`user.getPhotos()[0].isPrimary()`
	type	String	`user.getPhotos()[0].getType()`
	value	String	`user.getPhotos()[0].getValue()`
ims		List of ims	`ims = user.getIms();`
	display	String	`user.getIms()[0].getDisplay()`
	primary	Boolean	`user.getIms()[0].isPrimary()`
	type	String	`user.getIms()[0].getType()`
	value	String	`user.getIms()[0].getValue()`

Managed Systems Account Object Attributes for Inbound Transformation

You can modify or alter the incoming account attribute data by applying data transformation rules during the data ingestion phase into the Orchestrated system.

Syntax to Fetch Account Attributes for Inbound Data Transformation

The attribute details can be fetched using the syntax:

```
get<FieldName>()
```

Example 5-3 Retrieve the user's given name

```
account.getName().getGivenName()
```

Table 5-6 Managed Systems Account Attributes for the Inbound Data Transformation

Attribute	Sub Attribute	Data Type	Syntax
fullName		Reference	`account.getFullName()`
	formatted	String	`account.getFullName().getFormatted()`
	familyName	String	`account.getFullName().getFamilyName()`

Table 5-6 (Cont.) Managed Systems Account Attributes for the Inbound Data Transformation

Attribute	Sub Attribute	Data Type	Syntax
	givenName	String	`account.getFullName().getGivenName()`
	middleName	String	`account.getFullName().getMiddleName()`
	honorificPrefix	String	`account.getFullName().getHonorificPrefix()`
	honorificSuffix	String	`account.getFullName().getHonorificSuffix()`
userName		String	`account.getUserName()`
displayName		String	`account.getDisplayName()`
description		String	`account.getDescription()`
primaryEmail		String	`account.getPrimaryEmail()`
userType		String	`account.getUserType()`
title		String	`account.getTitle()`
status		String	`account.getStatus()`
accountType		String	`account.getAccountType()`
provisionedByMechanism		String	`account.getProvisionedByMechanism()`
provisionedOnDate		String	`account.getProvisionedOnDate()`
resourceName		String	`account.getResourceName()`
startDate		Long	`account.getStartDate()`
name		String	`account.getName()`
userLogin		String	`account.getUserLogin()`
resourcesId		String	`account.getResourcesId()`
compartmentId		String	`account.getCompartmentId()`
domainId		String	`account.getDomainId()`
domainOCID		String	`account.getDomainOCID()`
region		String	`account.getRegion()`

Table 5-6 (Cont.) Managed Systems Account Attributes for the Inbound Data Transformation

Attribute	Sub Attribute	Data Type	Syntax
emails		List of Email	`emails = account.getEmails()`
	pendingVerificationData	String	`account.getEmails()[0].getPendingVerificationData()`
	primary	Boolean	`account.getEmails()[0].getPrimary()`
	secondary	Boolean	`account.getEmails()[0].getSecondary()`
	type	String	`account.getEmails()[0].getType()`
	value	String	`account.getEmails()[0].getValue()`
	verified	Boolean	`account.getEmails()[0].getVerified()`
addresses		List of Address	`addresses = account.getAddresses();`
	country	String	`account.getAddresses()[0].getCountry()`
	formatted	String	`account.getAddresses()[0].getFormatted()`
	locality	String	`account.getAddresses()[0].getLocality()`
	postalCode	String	`account.getAddresses()[0].getPostalCode()`
	primary	Boolean	`account.getAddresses()[0].isPrimary()`
	region	String	`account.getAddresses()[0].getRegion()`
	streetAddress	String	`account.getAddresses()[0].getStreetAddress()`
	type	String	`account.getAddresses()[0].getType()`
phoneNumbers		List of PhoneNumber	`phoneNumbers = account.getPhoneNumbers()`
	display	String	`account.getPhoneNumbers()[0].getDisplay()`

Table 5-6 (Cont.) Managed Systems Account Attributes for the Inbound Data Transformation

Attribute	Sub Attribute	Data Type	Syntax
	primary	Boolean	`account.getPhoneNumbers()[0].isPrimary()`
	type	String	`account.getPhoneNumbers()[0].getType()`
	value	String	`account.getPhoneNumbers()[0].getValue()`
		Boolean	`account.getPhoneNumbers()[0].isVerified()`
photos		List of photos	`photos = account.getPhotos()`
	display	String	`account.getPhotos()[0].getDisplay()`
	primary	Boolean	`account.getPhotos()[0].isPrimary()`
	type	String	`account.getPhotos()[0].getType()`
	value	String	`account.getPhotos()[0].getValue()`
ims		List of ims	`ims = account.getIms()`
	display	String	`account.getIms()[0].getDisplay()`
	primary	Boolean	`account.getIms()[0].isPrimary()`
	type	String	`account.getIms()[0].getType()`
	value	String	`account.getIms()[0].getValue()`

Custom User and Account Attributes

You can fetch and use custom user or account attributes while applying data transformation rules for inbound data transformations. Outbound data transformations allow for fetching custom user attributes only.

User Custom Attribute

Oracle Access Governance provides a utility method to fetch the custom attribute of a user for inbound or outbound transformations. To fetch the CUSTOM_ATTRIBUTE_NAME of a user, you would use the following syntax, for example:

```
if( user.getCustomAttributes() != null ) {user.getCustomAttributes()
['CUSTOM_ATTRIBUTE_NAME'] }
```

For example, for a custom attribute called *Tags*:

```
if( user.getCustomAttributes() != null ) {user.getCustomAttributes()['Tags'] }
```

Account Custom Attribute

Oracle Access Governance provides a utility method to fetch the custom attribute of an account for inbound transformations only. To fetch the CUSTOM_ATTRIBUTE_NAME of an account, you would use the following syntax, for example:

```
if(account.getCustomAttributes() != null) {account.getCustomAttributes()
['CUSTOM_ATTRIBUTE_NAME'] }
```

For example, for a custom attribute called *Tags*:

```
if(account.getCustomAttributes() != null) {account.getCustomAttributes()
['Tags'] }
```

Examples for Outbound Data Transformation

Here are a few sample mapping rules and uses cases while applying outbound data transformations in Oracle Access Governance.

Table 5-7 Sample Mapping Rules

Usecase	Sample Rule
Fixed string value	`'SampleValue'`

Get user	Allows you to get the user keyed on the global identity id. This utility returns the user object from which you can then obtain other attributes.
	
	`transformationUtil.getUser(agcs_tenant_id, 'global_identity_id')`
	
	For example:
	
	`let user = transformationUtil.getUser(agcs_tenant_id, 'global_identity_id');`
	`if(user != null ) {`
	`  let username = user.getUserName();`
	`}`

Table 5-7 (Cont.) Sample Mapping Rules

Usecase	Sample Rule
User attribute	`user.getName().getGivenName()` **Note:** You must perform a null check before using such operations as the value can be null.

Date attribute	`new Date().getTime();`
	For example, to set the date to 31st Jan 2024:
	`new Date(2024,00,31).getTime();`

Application attribute	`application.getDisplayName()` **Note:** You must perform a null check before using such operations as the value can be null.
Request attribute	`requestAttributes.get('organizationName').get(0)` **Note:** • You must perform a null check before using such operations as the value can be null. • The 'requestAttributes .get(attrName)' always returns an array, so we need to do a get(i) to fetch the specific value

Table 5-7 (Cont.) Sample Mapping Rules

Usecase	Sample Rule

Set value to the combination of 2 user attributes	`user.getName().getGivenName() + ' ' + user.getName().getFamilyName()`
	
	or:
	
	`[ user.getName().getGivenName(),user.getName().getMiddleName() , user.getName().getFamilyName()].join( ' ')`

Set the value to another attribute if the input value is null (if organization is null then set to a fixed value)	`user.getOrganization() != null && user.getOrganization().getDisplayName() != null ? user.getOrganization().getDisplayName() : 'DefaultOrganization'`

Get code and decode value of a target lookup	`transformationUtil.getLookupCode(agcs_tenant_id, agcs_target_id, 'countries', user.getAddresses().get(0).getCountry() )`
	
	`transformationUtil.getLookupDecode(agcs_tenant_id, agcs_target_id, 'countries', user.getAddresses().get(0).getCountry() )`

Get code and decode value of a global lookup	`transformationUtil.getGlobalLookupCode( agcs_tenant_id, 'countries', user.getAddresses().get(0).getCountry() )`
	
	`transformationUtil.getGlobalLookupDecode(agcs_tenant_id, 'countries', user.getAddresses().get(0).getCountry() )`

Check if the User has Direct Reportee(s)	Check if the user has any direct reportees. This utility returns `TRUE` if the user has one or more reportees, otherwise it returns `FALSE`. For example: `transformationUtil.hasDirectReportees(agcs_tenant_id,'global_identity_id' )` .

Examples for Inbound Data Transformation and Identity Attributes

Here are a few sample mapping rules and uses cases while applying inbound data transformations or applying transformations on the composite identity profile in Oracle Access Governance.

> **Note:**
>
> As a best practice, we recommend to always perform a NULL check in rules for extracted values before using them otherwise it can lead to ingestion cycle failures on NULL references. This has to be done for both, *user* attributes object in Authoritative Sources and *account* attributes object in Managed Systems for inbound transformations.

Sample Mapping Rules for Authoritative Sources

Here are a few mapping rule expressions along with input value or output value for the identity (user) object attributes.

Target attribute	Type of attribute	Target attribute data type	Aim of mapping rule	Mapping rule expression	Value input	Value output
userName	DEFAULTS	String	Concatenate userName & displayName and set this value in userName attribute	`user.getUserName().concat('-', user.getDisplayName())`	userName=mark.hill displayName =Mark Hill	mark.hill-Mark Hill
userName	DEFAULTS	String	If userName is not null, then convert userName to upperCase and set in userName attribute	`if(user.getUserName()!=null) {user.getUserName().toUpperCase() }`	userName=mark.hill	MARK.HILL
jobDescription	CUSTOM	String	LowerCase the value of description and set it in custom attribute, jobDescription	`user.getDescription().toLowerCase()`	description = SoftwareDeveloper1	jobDescription = softwaredeveloper1
status	DEFAULTS	Boolean	If status is null set it to true else alternate the value.	`user.getStatus()==null ? true : !user.getStatus()`	status = true	false

Target attribute	Type of attribute	Target attribute data type	Aim of mapping rule	Mapping rule expression	Value input	Value output
risk	DEFAULTS	Integer	If risk is null set 20, else increase risk by 15	`user.getRisk() == null ? 20 : user.getRisk() + 15`	risk = 30 risk = null	45 20
description	DEFAULTS	String	Get startDate of type long, convert it into Date and then set it as a String to the description attribute.	`new Date(user.getStartDate()).toISOString()`	startDate = 1703442600000	2023-12-25T07:55:46.061Z
provisionedOnDate	DEFAULTS	Date	Get validFromDate (long), convert to date, then set provisionedOnDate rounded to 1st of next month.	`const currentDate = new Date(user.getValidFromDate()); new Date(currentDate.getFullYear(), currentDate.getMonth() + 1, 1).getTime();`	validFromDate = 1703442600000	provisionedOnDate = 1704047400000
provisionedFromDate	DEFAULTS	Date	Input type string. Output type date.	new Date(user.getValidFromDate()).toISOString()	validFromDate = 1703442600000 input = 2023-12-24T18:30:00.000Z	provisionedFromDate = 1703422800000

Sample Mapping Rules for Managed Systems

Here are a few mapping rule expressions along with input value or output value for the account object attributes.

Table 5-8 Sample Mapping Rules for Managing Permissions

Target attribute	Type of attribute	Target attribute data type	Aim of mapping rule	Mapping rule expression	Value input	Value output
displayName	DEFAULTS	String	If displayName is not null then set upper case value to displayName.	`if(account.getDisplayName()!=null){account.getDisplayName().toUpperCase()}`	displayName = Mark Hill	MARK HILL
primaryEmail	DEFAULTS	String	Concatenate userLogin & "@myexample.com" and set in primaryEmail.	`account.getUserLogin().concat('@myexample.com')`	userLogin = mark.hill	mark.hill@myexample.com
jobDescription	CUSTOM	String	LowerCase the value of description and set it in custom attributes jobDescription.	`if(account.getDescription()!=null){ account.getDescription().toLowerCase()}`	description = SoftwareDeveloper1	jobDescription = softwaredeveloper1
status	DEFAULTS	Boolean	Example 1: If status is null then set it to true else alternate the value.	`account.getStatus()==null ? true : !account.getStatus()`	status = true	false
			Example 2: Set status to false.	`false`	status = null/ true/false	false
risk	DEFAULTS	Integer	If risk is null then set to 20, else increase risk by 15.	`account.getRisk() == null ? 20 : account.getRisk() + 15`	risk = 30 risk = null	45 20
riskSummary	DEFAULTS	Long	If riskSummary is null set to 1234, else increase risk by 70.	`account.getRiskSummary() == null ? 1234 : account.getRiskSummary() + 70`	riskSummary = 30 riskSummary = null	100 1234

Oracle Access Governance Integration Functional Overview: Supported Operations in Orchestrated System

Oracle Access Governance enables integration with many native, direct or specialized applications and systems, either as an authoritative source or managed system.

This integration support allows you to manage use cases including configuration of orchestrated systems, data load, account creation and revocation, password change, and assignment and removal of roles.

Configure Orchestrated System

The first task you need to carry out to enable integration of your application or system with Oracle Access Governance is setup and configuration of an orchestrated system. This gives Oracle Access Governance details of how to connect to the target application or system from which you want to load data, or manage permissions. Optionally you can configure further elements of the Orchestrated System before running the initial dataload including:

- Notification Settings
- Identity/Account Matching Rules
- Apply data transformations to inbound and outbound data
- Identity attributes

Load Data

Once you have setup and verified your orchestrated system, you can run dataloads to ingest identity and account details, depending on the configuration mode you have selected, *Authoritative Source* or *Managed System*.

Data loaded in Authoritative Source mode will consist of user data from the orchestrated system. If the user is new, then a new identity is created in Oracle Access Governance. If the identity already exists in Oracle Access Governance, then any updates initiated in the orchestrated system will be applied.

Data loaded in Managed System mode comprises account data and permissions from the orchestrated system. If the account is provisioned from Oracle Access Governance, then a new account is created, together with associated permissions, in the orchestrated system. Accounts and permissions directly loaded from your orchestrated system can be managed by Oracle Access Governance. You can remediate permissions associated with a managed system account. If the account only has one permission assigned then remediation of this permission will also result in the revoking of the account.

Create Account

An account can be created in Oracle Access Governance in two ways:

- Ingesting account data from your orchestrated system.
- When a role, policy, or access bundle containing application permissions is assigned to an identity. If you have an identity in Oracle Access Governance then you can request an account by using the **Request a new access** functionality in the Oracle Access

Governance console. If you make an access request for an access bundle or permission which is approved, a provisioning operation will be initiated. The provisioning process will, if there is no existing account managed by Oracle Access Governance, create an account on the chosen application. If an account managed by Oracle Access Governance already exists, then the permissions for that account are updated based on the values in the access bundle.

For further details about account creation, refer to Request Access.

Assign Permissions

You can assign permissions to an account using the **Request a new access** functionality of Oracle Access Governance. This allows you to request an access bundle containing permissions applicable to your application. When you request an access bundle, either directly or through an Oracle Access Governance role or policy, a provisioning operation is initiated which updates the permissions in your application with the permissions included in the referenced access bundle.

For further details about permission assignment, refer to Request Access. To learn more about roles and policies, refer to Manage Roles, and Manage Policies.

Remove Permissions

You can remove permissions from an account by revoking the permission from the role, policy or access bundle to which it is assigned. In this case, the permission assignment is revoked from all users to whom the role, policy or access bundle is applied. Say you had an access bundle with two permissions, *Admin*, and *Developer* which had previously been provisioned to your application. You could update the access bundle containing these permissions to remove *Developer* and add *Composer*, resulting in the access bundle containing *Admin*, and *Composer*. This change would be reflected following the next provisioning operation, by removing the *Developer* role and assigning the *Composer* role. *Admin* would remain assigned.

Another way to remove a permission would be by revoking role, policy or access bundle assignment from a specific account. This would be done using the revoke operation in access reviews.

For further details about permission assignment, refer to Delete a Role, Delete a Policy, or Manage Access Bundles -> Delete an Access Bundle.

Users with the `AG_ServiceDesk_Admin` role can directly revoke permissions from the **Manage Identities** page, using the **Revoke permission** operation. The Grant Type of these permissions must either be `DIRECT` or Access Bundles granted through `REQUEST`. You cannot revoke permissions for Oracle Cloud Infrastructure (OCI) or Oracle Identity Governance (OIG) systems. For detailed steps, see Revoke one or multiple permissions for an Account.

Change Password

The ability to change an account password is provided by the **My Access** functionality in Oracle Access Governance Console. If you change the account password in this page, the details will be sent to the chosen application in the next provisioning operation, and the password change is applied to your Database Application Tables (Oracle) account.

For further details about changing passwords, refer to Change Account Password.

Revoke Account

If you revoke an account in an access review, provisioning tasks will be created to revoke the account in the corresponding application. For further details about revoking accounts, refer to Delete a Role, or Delete a Policy.

Users with the `AG_ServiceDesk_Admin` role can now directly disable accounts managed by Oracle Access Governance from the **Manage Identities** page, using the **Disable account** operation. Once disabled all the associated accesses are revoked. The accounts can still be managed by Oracle Access Governance. For detailed steps, see Disable and Enable an Account Managed by Oracle Access Governance.

You may delete accounts using the **Delete account** operation. For deleted accounts, all the associated accesses are removed and you can no longer manage the accounts from Oracle Access Governance. For detailed steps, see Delete an Account Managed by Oracle Access Governance.

Enable Account

Users with the `AG_ServiceDesk_Admin` role can re-provision the accounts and the accesses using the **Enable account** operation from the **Manage Identities** page. Once enabled, all the accounts and accesses are re-provisioned, into Oracle Access Governance. For detailed steps, see Disable or Enable an Account Managed by Oracle Access Governance.

Integrate with Orchestrated Systems

Preinstall

Certified Components

The target system can be any one of the following:

- Oracle Identity Governance **12.2.1.4 Bundle Patch Number 11 (12.2.1.4.220703) or later**. If your current version of Oracle Identity Governance is not compatible then contact Oracle Support, who can arrange a patch for your Oracle Identity Governance system
- Oracle Identity Governance **14.1.2.1.0 or later**.

Prerequisites

The Oracle Identity Governance source data must meet the following requirements to be eligible for review in Oracle Access Governance:

- Applications and Entitlements in Oracle Identity Governance must be marked as **Certifiable** in order to be ingested by Oracle Access Governance. Log in to the Oracle Identity Governance Self Service application and navigate to **Request Access → Request for Self → [Search for Your App]** and click the information icon, and select the **Certifiable** flag.

- For Roles, log in to the Oracle Identity Governance Self Service application and navigate to **Manage → Roles → Open the Role**. Under Catalog Attributes, select the **Certifiable** check box.
- Any access included in an Oracle Access Governance review must have been granted using one of the following grant types in Oracle Identity Governance:
  - **Direct Provision accounts and Entitlements**
  - **Request Provision accounts and Entitlements**
  - **Reconciled accounts and Entitlements from the targets**
  - **Bulkloaded accounts and Entitlements**
  - **Request or Direct provision Role which are associated with access policy**

Set Up Oracle Identity Governance Integration

To enable the Oracle Identity Governance agent to connect to Oracle Access Governance, you need to enter connection details and credentials for the target system, and build an agent specific to your environment.

1. In a browser, navigate to the Oracle Access Governance service home page and log in as a user with the **Administrator** application role.
2. On the Oracle Access Governance service home page, click on the  icon and select **Service Administration** and then **Orchestrated Systems**.
3. Select the **Add an orchestrated system** button, to navigate to the **Add an orchestrated system** page to start the workflow.
4. On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the Search field.
   a. Select **Oracle Identity Governance**
   b. Select **Next**
5. On the **Enter Details** step, enter the general details for the orchestrated system:
   - Enter a name for the system you want to connect to in the **What do you want to call this application?** field.
   - Enter a description for the system in the **How do you want to describe this application?** field.
   - Click **Next**.
6. On the **Add owners** step:

   You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

a. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.

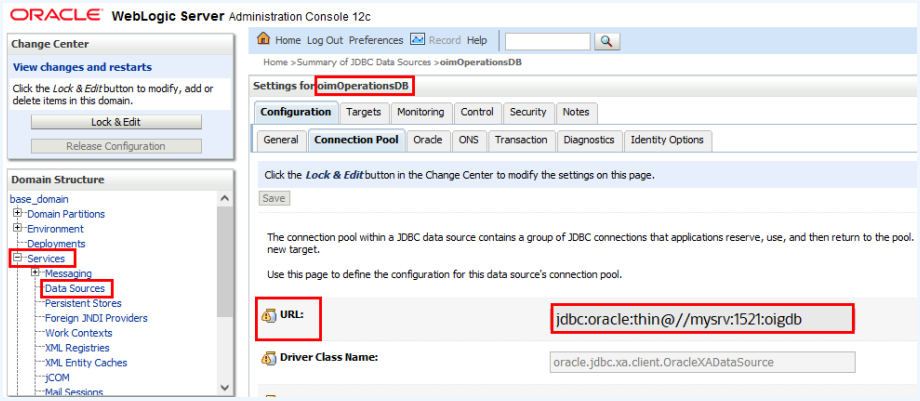
b. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

7. On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the required Oracle Identity Governance instance.

- In the **What is the JDBC URL of your OIG database server?** field, enter the JDBC URL for the OIG database you want to connect with.

> **Note:**
>
> To obtain the JDBC URL:
>
> a. Log on to the Oracle WebLogic Server Administration Console associated with your Oracle Identity Governance instance.
>
> b. Navigate to **Services** → **Data Sources**.
>
> c. Select **oimOperationsDB** from the **Configurations** tab.
>
> d. Select **Connections Pool**, and copy the value from the **URL:** field to use as the JDBC URL for Oracle Identity Governance.

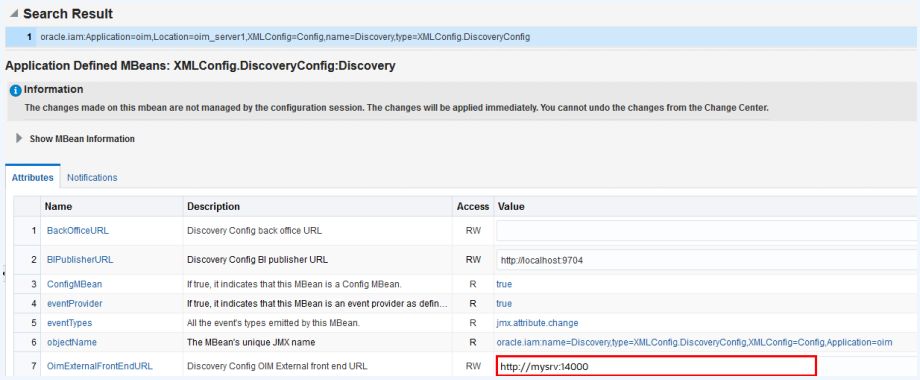


- In the **What is the OIG database user name?** field, enter the database user to connect to the OIG database.

> **Note:**
>
> This can be any user with read access to the OIG database.

- In the **Password** field, enter the password for the OIG database user you have specified.
- In the **What is the URL of your OIG server?** field, enter the URL of the OIG server you want to integrate with.

> **Note:**
>
> To obtain the OIG Server URL:
>
> – Log on to the Oracle Enterprise Manager Fusion Middleware Control.
>
> – Navigate to the **System MBean Browser** and locate the **XMLConfig.DiscoveryConfig** MBean.
>
> – Copy the value of the **OimExternalFrontEndURL** attribute and use this as the value for the Oracle Identity Governance Server URL.



- In the **What is the OIG server user name?** field, enter the OIG user used for remediation and schema discovery.

> **Note:**
>
> The Oracle Identity Governance Server user can be any user that is a member of the **System Administrator** administration role. This role is required to perform the remediation process, and to support schema discovery for custom attributes. In the case where only remediation support is needed then user can be a member of the **OrclOAGIntegrationAdmin** administration role. With this user the schema discovery operation will fail.

- In the **OIG server password** field, enter to authenticate the OIG server user when calling OIG APIs to perform remediation.

> **Note:**
>
> Information about the Oracle Identity Governance Server (URL, Username, and Password), and Oracle Identity Governance datasource (JDBC URL, Username, and Password) is required to integrate Oracle Access Governance and Oracle Identity Governance. Oracle Access Governance will use the Oracle Identity Governance data source to load the data and the Oracle Identity Governance Server URL to perform remediation operations. In case of a connection failure, the Oracle Access Governance agent automatically retries a maximum of three times to secure a connection with the Oracle Identity Governance server.

8. Optionally, you can select to perform data loads using OIG database incremental data load. If you select the **Do you want to enable OIG database incremental data load?** option then Day-N data loads will use an event-driven mode which applies changes to Oracle Access Governance as they happen, rather than as a periodic snapshot. If you select this option, ensure that you have completed the prerequisite tasks in the OIG database defined in Database Setup Steps for Event-driven Data Load.

   > **Note:**
   >
   > You should use this option if you want to see events from OIG in real-time rather than periodically. For example, if your organization creates an identity for a user which needs to be reflected in Oracle Access Governance immediately then you should use this option. When the identity is added, the event is noted by the integration and reconciled with AG. The default snapshot data load will not reconcile the new identity until its next scheduled run, when it will run a data load of all changes since the last. With the event-driven data load, changes are identified in real-time and loaded into Oracle Access Governance as each event takes place.

   .

9. Enter filter attributes which will be used to filter the data that is returned from OIG.

   You can add up to three filter name/value pairs which will be used to restrict the users and accounts ingested from Oracle Identity Governance by Oracle Access Governance. You can also set the search filter values separator to a character of your choice if required (default is ~).
   Details of the attributes you can use to set filters against can be found in Supported Attributes for User Data Load Filtering.

10. Verify the details entered are correct, and click the **Add** button

11. On the **Download Agent** step, select the **Download** link and download the agent zip file to the environment in which the agent will run.

    After downloading the agent, follow the instructions explained in the Agent Administration article.

12. You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

    - **Customize before enabling the system for data loads**

- **Activate and prepare the data load with the provided defaults**: If you select this option, the default matching rule for **Oracle Identity Governance** orchestrated system will be used.

Table 5-9 Default Matching Rule

Mode	
Authoritative Source	`userName = userName`
Managed System	Default rule is not supported. Matching is based on UID.

You can also follow the instructions provided in the Set Up Identity Orchestration between Oracle Access Governance and Oracle Identity Governance (OIG) tutorial.

Supported Attributes for User Data Load Filtering

When configuring an Orchestrated System to on-board data from Oracle Identity Governance, it is possible to filter the user data you want to ingest in Oracle Access Governance. You can restrict which users are on-boarded by setting filters on identity attributes such as department, employee type, location, and others.

User Data Load Filtering Characteristics

You should be aware of the following characteristics of user data load filtering befire configuring filters in your Orchestrated System.

- Matching of user search filters and user data values filtering is case sensitive. For example. a filter of `department = Human Resources` would not return users with a value of `department = HUMAN RESOURCES`, or `Department = Human Resources`.
- If no users or accounts match the user data load filter, then no data will be ingested from Oracle Identity Governance by Oracle Access Governance. In this case, however, the data load itself will be labelled as successful in the activity log, even though no identities or accounts are on-boarded.
- User data load filter values cannot exceed 1000 for any given filter attribute.
- If an agent is already installed, an agent upgrade is required to enable user data load filters. See Agent Example Usage for details on how to upgrade your agent.

List of Supported Attributes for User Data Load Filtering

You can filter users ingested from Oracle Identity Governance based on the following attributes.

Table 5-10 List of Supported Attributes for User Data Load Filtering

Oracle Access Governance Attribute Name	Oracle Identity Governance Attribute Name
employeeType	usr_emp_type
jobCode	usr_job_code
department	usr_dept_no
location	usr_location
state	usr_state
postalCode	usr_postal_code
country	usr_country

Table 5-10 (Cont.) List of Supported Attributes for User Data Load Filtering

Oracle Access Governance Attribute Name	Oracle Identity Governance Attribute Name
managerUid	usr_manager_key
managerLogin	usr_login (usr_login of manager)
organizationUid	act_key
organizationName	act_name act_name of act table
territory	usr_territory

Example User Data Load Filters

Some examples of usecases you can configure using the User Data Load Filter functionality are provided below:

Table 5-11 Example User Data Load Filters

Usecase	Configuration Parameters
User with **department=Product Development** and **jobCode=IC004 or M0003**	• userFilter1Name=department • userFilter1Value=Product Development • userFilter2Name=jobCode • userFilter2Value=IC004~M0003 • userFilter3Name= • userFilter3Value= • filterValueDelimiter=~
User with **state =Kent** and **organizationUid=1 or 4**	• userFilter1Name=state • userFilter1Value=Kent • userFilter2Name=organizationUid • userFilter2Value=1~4 • userFilter3Name= • userFilter3Value= • filterValueDelimiter=~
User with **postalCode = 78045 or 12204** with **custom delimiter ##**	• userFilter1Name=postalCode • userFilter1Value=78045##12204 • userFilter2Name= • userFilter2Value= • userFilter3Name= • userFilter3Value= • filterValueDelimiter=##
User with **managerUid = 17981 or 17854** and **managerLogin = DINORAH.PREWITT or JOELLA.SHANNON**	• userFilter1Name=managerUid • userFilter1Value=17981~17854 • userFilter2Name=managerLogin • userFilter2Value=DINORAH.PREWITT~SHIRLEY.THOMAS • userFilter3Name= • userFilter3Value= • filterValueDelimiter=~

Note:

Filter value name and the value of the filter are both case sensitive. Using the example above, any of the following would be an invalid filter, and return no results:

- Example 1:
  - userFilter1Name=**MANAGERUID**
  - userFilter1Value=17981~17854
  - userFilter2Name=managerLogin
  - userFilter2Value=DINORAH.PREWITT~SHIRLEY.THOMAS
- Example 2:
  - userFilter1Name=managerUid
  - userFilter1Value=17981~17854
  - userFilter2Name=managerLogin
  - userFilter2Value=**Dinorah.Prewitt**~SHIRLEY.THOMAS
  - Example 3:
    - **USERFilter1Name**=managerUid
    - userFilter1Value=17981~17854
    - userFilter2Name=managerLogin
    - userFilter2Value=DINORAH.PREWITT~SHIRLEY.THOMAS
- Example 4:
  - userFilter1Name=managerUid
  - userFilter1Value=17981~17854
  - userFilter2Name=managerLogin
  - **USERFILTER2VALUE**=DINORAH.PREWITT~SHIRLEY.THOMAS

Database Setup Steps for Event-driven Data Load

When creating or updating an Oracle Access Governance orchestrated system you can enable the event-driven data load option. This option switches Day-N data load from the default snapshot-based model, to an event-driven one. A prerequisite for this option requires you to create a read-only user in the OIG database and grant required roles and system privileges.

To add a read-only user in the OIG database for the event-driven data load option, complete the following steps:

1. Connect to the OIG database as SYS and create a read-only user in the OIG database that will be used by Oracle Access Governance to connect to access change events:

   ```
   create user <username> identified by <password>;
   ```

   For example:

   ```
   create user ag2oigro identified by mypassword;
   ```

2. Connect to the OIG database as SYS and grant the required roles and system privileges to the read-only user you created in the previous step:

```
GRANT CREATE SESSION TO <read-only user>;
GRANT SELECT ANY TABLE TO <read-only user>;
GRANT CREATE ANY TRIGGER TO <read-only user>;
GRANT ADMINISTER DATABASE TRIGGER TO <read-only user>;
GRANT CREATE TABLE TO <read-only user>;
GRANT CREATE SYNONYM TO <read-only user>;
GRANT UNLIMITED TABLESPACE TO <read-only user>;

GRANT CONNECT TO <read-only user>;
GRANT RESOURCE TO <read-only user>;
```

For example:

```
GRANT CREATE SESSION TO ag2oigro;
GRANT SELECT ANY TABLE TO ag2oigro;
GRANT CREATE ANY TRIGGER TO ag2oigro;
GRANT ADMINISTER DATABASE TRIGGER TO ag2oigro;
GRANT CREATE TABLE TO ag2oigro;
GRANT CREATE SYNONYM TO ag2oigro;
GRANT UNLIMITED TABLESPACE TO ag2oigro;

GRANT CONNECT TO ag2oigro;
GRANT RESOURCE TO ag2oigro;
```

3. Connect to the OIG database as the OIG DB Schema Owner and run the following command to create a script that will create synonyms for OIG tables for the read-only user:

```
setheading on
setlinesize 1500
setnumformat 99999999999999999999
setpagesize 25000
spool synon.out
SELECT 'create synonym <read-only user>.'||TNAME||' for
<OIG_SCHEMA_USER_NAME>.'||TNAME||';'
FROM TAB
WHERE tabtype = 'TABLE';
spool off
```

For example:

```
setheading on
setlinesize 1500
setnumformat 99999999999999999999
setpagesize 25000
spool synon.sql
SELECT 'create synonym ag2oigro.'||TNAME||' for <OIG_SCHEMA_USER_NAME>.'||
TNAME||';'
FROM TAB
WHERE tabtype = 'TABLE';
spool off
```

4. Connect to the OIG database as the read-only user, and create the synonyms using the script created in the previous step:

   ```
   @<scriptname>
   ```

   For example:

   ```
   @synon.sql
   ```

Configure Integration Between Oracle Access Governance and Oracle Cloud Infrastructure (OCI)

You can establish a connection between Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) and Oracle Access Governance as an authoritative source and as a managed system. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

The OCI Orchestrated System supports the following modes and both included, by default:

- **Authoritative Source**: You can use OCI as an authoritative (trusted) source of identity information for Oracle Access Governance.
- **Managed System**: You can manage OCI IAM groups and application roles, including certifying policies and group membership using Oracle Access Governance. See Supported Operations for Oracle Cloud Infrastructure (OCI).

Set Up Identity Resources on OCI to Connect to Oracle Access Governance

Before you can establish a connection, you need to create, manage, and provision identity resources in your cloud tenancy.

Prerequisites

The following prerequisites must be satisfied to integrate with Oracle Access Governance with OCI IAM:

- Your cloud account must use Identity Domains to manage identities on OCI.
- As a cloud administrator, you must be able to manage identities in the **Default** domain and manage policies in the **root compartment** of your tenancy.

You must set up the following in your cloud tenancy:

- Create a local identity user: *agcs_user* in the **Default** domain of your root compartment.
- Create an identity group: *agcs_group* in the **Default** domain of your root compartment.
- Edit user capabilities and select **API keys** for *agcs_user*
- Assign identity user *agcs_user* to the identity group *agcs_group*
- Add the following group policies in tenancy:

  ```
  allow group agcs_group to inspect all-resources in tenancy
  allow group agcs_group to manage policies in tenancy where all {request.permission
  in ('POLICY_UPDATE', 'POLICY_DELETE'), target.policy.name !='Tenant Admin Policy'}
  allow group agcs_group to manage domains in tenancy
  allow group agcs_group to read audit-events in tenancy
  allow group agcs_group to read objectstorage-namespace in tenancy
  ```

You can either manually perform these actions on OCI, or use a Terraform script to automatically set up these identity resources.

> **Note:**
>
> We recommend using the Terraform script to create and manage your identity resources and avoid any technical glitches or errors during the setup.

After you have set up the above mentioned identity resources, you need to generate API Keys for the identity user, and make a note of the Oracle Cloud Identifier (OCID) which you will use to configure your cloud provider.

Set up Identity Resources Manually

Follow the steps and the links provided to set up identity resources manually in your cloud tenancy.

1. Create an identity user, *agcs_user*, in the **Default** domain for Oracle Access Governance access.
2. Provision the user with the following capabilities:
   - API keys: Select the check box for API authentication.
3. Create an identity group, *agcs_group*, in the **Default** domain for Oracle Access Governance API access and user assignment.
4. Assign the identity user (*agcs_user*) to the identity group (*agcs_group*)
5. Create the following policies for the identity group in the root compartment:

```
allow group agcs_group to inspect all-resources in tenancy
allow group agcs_group to manage policies in tenancy where all {request.permission
in ('POLICY_UPDATE', 'POLICY_DELETE'), target.policy.name !='Tenant Admin Policy'}
allow group agcs_group to manage domains in tenancy
allow group agcs_group to read audit-events in tenancy
```

Set up Identity Resources using a Script

Use a Terraform script to automatically create, manage, and provision the identity resources. Perform the following tasks for running the Terraform script:

1. Create a Terraform Script File
   a. Copy the following script in a text editor and save it with the *.tf* file extension.

```
variable "compartment_ocid" {}
variable "region" {}

provider "oci" {
  region = var.region
}

#Identity group
resource "oci_identity_group" "agcs_group" {

  compartment_id = var.compartment_ocid
  description    = "Group for AGCS API Access"
  name           = "agcs_group"
```

```
  freeform_tags = { "AGCS" = "true" }
}

#AGCS User
resource "oci_identity_user" "agcs_user" {

  compartment_id = var.compartment_ocid
  description    = "Local User for AGCS access"
  name           = "agcs_user"
  email          = <Set your email address>

  freeform_tags = { "AGCS" = "true" }
}

#AGCS User Capabilities
resource "oci_identity_user_capabilities_management"
"agcs_user_capabilities_management" {

  user_id                      = oci_identity_user.agcs_user.id
  can_use_api_keys             = "true"
  can_use_auth_tokens          = "false"
  can_use_console_password     = "false"
  can_use_customer_secret_keys = "false"
  can_use_smtp_credentials     = "false"
}

# AGCS Policy
resource "oci_identity_policy" "agcs_policy" {
  compartment_id = var.compartment_ocid
  description    = "AGCS Policy"
  name           = "agcs_policy"
  statements = ["allow group agcs_group to inspect all-resources in
tenancy
allow group agcs_group to manage policies in tenancy where all
{request.permission in ('POLICY_UPDATE', 'POLICY_DELETE'),
target.policy.name !='Tenant Admin Policy'}
allow group agcs_group to manage domains in tenancy
allow group agcs_group to read audit-events in tenancy"]

  freeform_tags = { "AGCS" = "true" }
}

# Group assignment
resource "oci_identity_user_group_membership" "agcs_group" {
  group_id = oci_identity_group.agcs_group.id
  user_id  = oci_identity_user.agcs_user.id
}
```

b. Update the cloud administrator email address in the `email` field to a valid email address and enable creation of the identity user.

c. Replace any other value that you may want to update.

d. Compress the file folder and save it with the *.zip* file extension.

2. **Use the Stack Service of OCI Resource Manager to run Terraform Script**

Use OCI Resource Manager to run the Terraform script. In this task, you will:

a. Sign in to https://cloud.oracle.com. using your Cloud Administrator credentials.

b. When you have successfully logged in, select region depending on your home region location. For example, from the top navigation menu, select **US East (Ashburn)**.

c. Open the navigation menu icon and select **Developer Services**.

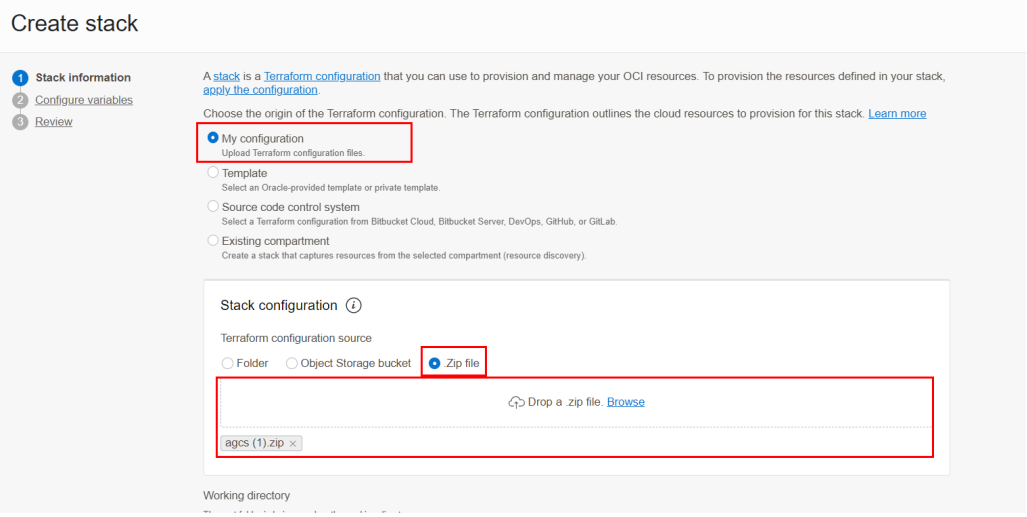
d. Under **Resource Manager**, click **Stacks**.

e. On the left pane, choose a compartment.

f. Click **Create Stack**. The **Create Stack** page is displayed with the **Stack Information** tab opened by default.

g. Select **My Configuration**.

h. In **Stack Configuration**, select the **.Zip file** option, and then drag and drop your terraform zip file in the marked space.



i. Enter the stack name and its description. Then, select the compartment.

j. Enter **Terraform version** as **1.0.x** and then click **Next**.

k. If applicable, review the **compartment_ocid** and **region** variable values.

l. Click **Next** to review the stack configuration,

m. To automatically provision resources on creation of the stack, select **Run apply**, and then click **Create**. The **Stack details** page is displayed.

3. **Parse the Script and Preview the Expected Output**

a. On the **Stack details** page, click **Plan**.

b. Modify the job name, if required, and then click **Plan**. The **Job details** page is displayed with the **Accepted state**. Wait for a few minutes till the job displays the **Succeeded** state.

c. You can view the logs indicating the set of actions that this terraform script will perform. Alternatively, you can download the logs file.

```
Terraform will perform the following actions:

  # oci_identity_group.agcs_group will be created
  + resource "oci_identity_group" "agcs_group" {
      + compartment_id = "ocid1.tenancy.oc1..aaaaaaaambnn3yrfpb55k7t6h2k3ptrf7es5jwb3ytyykojyf4dq2cd2z2xq"
      + defined_tags   = (known after apply)
      + description    = "Group for AGCS API Access"
      + freeform_tags  = {
          + "AGCS" = "true"
        }
      + id             = (known after apply)
      + inactive_state = (known after apply)
      + name           = "agcs_group"
      + state          = (known after apply)
      + time_created   = (known after apply)
    }

  # oci_identity_policy.agcs_policy will be created
  + resource "oci_identity_policy" "agcs_policy" {
      + ETag           = (known after apply)
      + compartment_id = "ocid1.tenancy.oc1..aaaaaaaambnn3yrfpb55k7t6h2k3ptrf7es5jwb3ytyykojyf4dq2cd2z2xq"
      + defined_tags   = (known after apply)
      + description    = "AGCS Policy"
      + freeform_tags  = {
          + "AGCS" = "true"
```

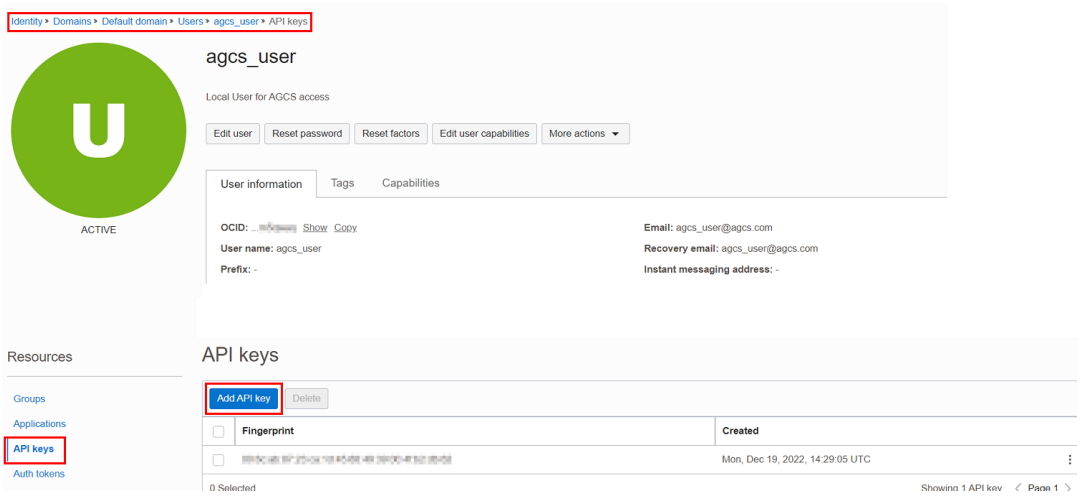
d. From the navigation menu, go to **Identity & Security**, and check the newly created resources in the Default domain. For example,

- *agcs_policy* under **Policies**
- *agcs_user* under **Domains** → **Default** domain → **Users**
- *agcs_group* under **Domains** → **Default** domain → **Groups**
- Within *agcs_group*, the *agcs_user* is assigned to that group

Generate API Keys and Oracle Cloud Identifier (OCID) to configure your Cloud Environment in the Oracle Access Governance Console

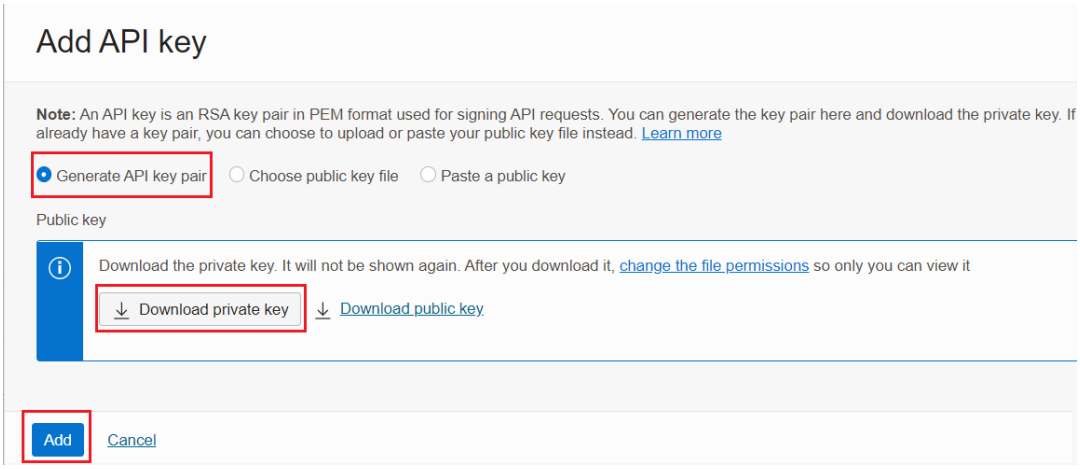
After you have set up the identity resources, you need to generate API Keys for the identity user (*agcs_user*) and note OCID for that identity user. You will use it to configure your cloud environment on the Oracle Access Governance Console.

1. In OCI console, from the navigation menu, select **Identity & Security**, and then **Domains** → your compartment, and then from the left pane, select **Users**.
2. Select the *agcs_user* that the script automatically created.
3. On the left pane, in the **Resources** section, select **API keys**.



4. Select **Add API key**, and then select **Generate API key pair** .
5. Download the private key and save it.

6. Click **Add**. The configuration file is created displaying *ocid*, *fingerprint*, *tenancy* and *region* details. Save the information available on the configuration file in a separate text file.



Establish Connection by Adding a New Orchestrated System - OCI IAM

Integration with Oracle Cloud Infrastructure Identity and Access Management (OCI IAM) is achieved by configuring a new orchestrated system with the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon, select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of application you would like to onboard.

1. Select **Oracle Cloud Infrastructure**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the system.

> **Note:**
>
> If you used the method explained in the Generate API Keys and Oracle Cloud Identifier (OCID) to configure your Cloud Environment in the Oracle Access Governance Console task to generate API keys and OCID for a user (*agcs_user*), then directly enter the saved values.

1. **What is the OCI user's OCID?**: Enter the Oracle Cloud Identifier (OCID) for the OCI user you will use to connect to the system. For further information regarding OCIDs see Oracle Cloud Identifier, OCID Syntax, and Where to Get the Tenancy's OCID and User's OCID. For example,
   `ocid1.user.oc1..aabdgsegsccawmw2o6qraopae7egmlochlopclhnwxq6pctu6oocgn`
2. **What is the OCI user's fingerprint?**: Enter the fingerprint of the public key of the API Signing Key for the OCI instance you will be connecting to. Steps to retrieve the fingerprint can be found in How to Get the Key's Fingerprint, The fingerprint will look similar to this:
   `12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef`.
3. **What is the OCI user's private SSH key?**: Enter the private SSH key (.pem file) for the API Signing Key. Copy it directly from the text editor or use the *cat* command to open the SSH key file from console.
4. **What is the OCI tenancy OCID?**: Enter the OCID for the target tenancy. For further information regarding OCIDs see Oracle Cloud Identifier, OCID Syntax, and Where to Get the Tenancy's OCID and User's OCID.

5. **What is the OCI tenancy's home region?**: Enter the home region for the target OCI tenancy, using the region identifier. The region identifier for your home region can be found in Regions, the identifier for US East (Ashburn) is `us-ashburn-1`, for example. For further information on home region, see The Home Region, and How do I find my tenancy home region?.

> **Note:**
>
> You cannot create multiple orchestrated systems using the same tenancy ID. Use a unique tenancy for each system.

6. **Which domain names should be included?**: By default, all domains in the tenancy will be ingested when you run a data load into Oracle Access Governance. If you have multiple domains you may want to restrict which domains are loaded. This parameter allows you to specify which domains you want to load. Select the domains from which you want to ingest data. If configured, you must include the service account user domain. If this is left blank, then all domains will be included.
7. Click **Add**.

Finish Up

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Supported Operations for OCI

The Oracle Cloud Infrastructure Orchestrated System supports the following account operations when provisioning a user.

Group Provisioning - Assign Groups to Users

You can assign multiple OCI IAM groups for an OCI domain from Oracle Access Governance. For example, you can provision developers working on multiple projects to different OCI IAM resources, each group associated with specific resources.

To do this:

- Create an access bundle for the OCI orchestrated system and select the OCI IAM groups available in the domain. For details, see Create Access Bundle.
- To assign users in OCI with OCI groups:
  - Create an Oracle Access Governance policy and associate the groups part of the access bundle with identity collection in that policy. For details, see Manage Policies and Create Identity Collections.
  - You may also request access bundles or roles directly by raising a request from the self service flows. For details, see Request Access.

You may certify identity access reviews for the groups granted through access request by Oracle Access Governance. For more information, see Eligible System Oracle Cloud Infrastructure.

Application Role Provisioning - Assign Roles

Using Oracle Access Governance, you can provision OCI application roles to OCI identities for services running in an OCI domain. You can even use this to provision Oracle Access Governance roles to other identities. For example, you can package relevant Oracle Access Governance application roles and provision the access bundle to an IAM Specialist group.

- Create an access bundle for the OCI orchestrated system and select application roles for services available in the domain. For details, see Create Access Bundle.
- To assign users in OCI with application roles:
  - Create an Oracle Access Governance policy. Associate access bundles comprising application roles with an identity collection in that policy. For details, see Manage Policies and Create Identity Collections.
  - You may also request this access bundle or role directly by raising a request from the self service flows. For details, see Request Access.

You may further certify identity access reviews for the roles granted through access request by Oracle Access Governance. For more information, see Eligible System Oracle Cloud Infrastructure.

OCI Policy Reviews : Revoke Over-privileged Policy Statements from OCI Policies

Using Oracle Access Governance, you can certify OCI policies by creating on-demand policy review campaigns from the Oracle Access Governance Console. For example, you may run quarterly reviews on the defined network and storage policy of your tenancy to assess if these meet the principle of least privilege and applicable regulatory requirements.

Create a policy review campaign for OCI policies. Based on the prescriptive insights and recommendations, reviewers can make informed decision to either **Approve** or **Reject** entire policy at once, or make decision to **Approve** or **Reject** specific policy statement in that policy.

For more information, see Review Access to Systems Managed by Oracle Cloud Infrastructure (OCI) and Create Policy Review Campaigns.

Group Membership Reviews: Accept or Revoke Membership Access from OCI IAM Group

Using Oracle Access Governance, you can certify membership for OCI IAM groups by creating on-demand Identity Collection Review campaigns. For example, you may run group membership reviews to certify that only eligible members are part of the *Database Administrator* group, managing and maintaining the database infrastructure for your project. An identity with the *Sales Analyst* role should not be associated with this group.

Create an identity collection review campaign for OCI IAM Groups. Based on the prescriptive insights and recommendations, reviewers can make informed decision to either **Approve** or **Revoke** members of the group. If you choose to review OCI IAM groups and it contains a few members provisioned from **Oracle Access Governance**, then with this review, you can only accept or revoke directly assigned members. For members provisioned from Oracle Access Governance, choose to review the OCI Access Bundles using the **Which permissions?** tile. For more information, see Review Access to Systems Managed by Oracle Cloud Infrastructure (OCI) and Create Identity Collection Review Campaigns.

The Database User Management connector integrates Oracle Access Governance with database user management tables in Oracle Database. You can establish a connection

between Oracle Database and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Prerequisites

Before you install and configure a Database User Management (Oracle) orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

You can integrate any one of the following Oracle Database types with Oracle Access Governance:

- **Exadata V2**.
- **Oracle Database 12c** as single database, pluggable database (PDB), or Oracle RAC implementation.
- **Oracle Database 18c** as single database, pluggable database (PDB), or Oracle RAC implementation.
- **Oracle Database 19c** as single database, pluggable database (PDB), or Oracle RAC implementation.
- **Oracle Database 23ai** as single database, pluggable database (PDB), or Oracle RAC implementation.
- **Oracle Autonomous Database**

Supported Operations

The Database User Management (Oracle) orchestrated system supports the following operations:

- Create user
- Reset password
- Add roles
- Revoke Roles
- Add privileges
- Revoke privileges

Default Supported Attributes

The Database User Management (Oracle) orchestrated system supports the following default attributes.

Table 5-12 Default Attributes - Manage Permission Mode

DBUM User Entity	Target Account Attribute	Oracle Access Governance Account Attribute
	Return Id	uid
	Username	name
	Authentication Type	authenticationType
	Global DN	globalDN
	Default Tablespace	defaultTablespace

Table 5-12 (Cont.) Default Attributes - Manage Permission Mode

DBUM User Entity	Target Account Attribute	Oracle Access Governance Account Attribute
	Default Tablespace Quota (in MB)	defaultTablespaceQuotaInMB
	Temporary Tablespace	temporaryTablespace
	Profile Name	profileName
	Account Status	accountStatus
	Status	status
	Password	password
Role DBUM (Oracle) roles are mapped to Oracle Access Governance entitlements		
	adminOption	roleAdminOption
Privilege DBUM (Oracle) privileges are mapped to Oracle Access Governance entitlements		
	adminOption	privilegeAdminOption

Default Matching Rule

The default matching rule for Database User Management (Oracle) orchestrated system

is:

Table 5-13 Default Matching Rules

Mode	Default Matching Rule
Manage Permissions	`userNameOracle = userLogin`

Create a Target System User Account for Database User Management (Oracle) Orchestrated System Operations

Oracle Access Governance requires a user account to access the system during service operations. Depending on the system you are using, you can create the user, and assign specific permissions and roles to them.

For Oracle database:

1. Create a service user using the following SQL statement:

```
CREATE USER agserviceuser IDENTIFIED BY password
DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp QUOTA UNLIMITED ON users;
```

2. Assign the following permissions and roles to the service user created:

```
GRANT SELECT on dba_role_privs TO agserviceuser;
GRANT SELECT on dba_sys_privs TO agserviceuser;
GRANT SELECT on dba_ts_quotas TO agserviceuser;
GRANT SELECT on dba_tablespaces TO agserviceuser;
GRANT SELECT on dba_users TO agserviceuser;
GRANT CREATE USER TO agserviceuser;
```

```
GRANT ALTER ANY TABLE TO agserviceuser;
GRANT GRANT ANY PRIVILEGE TO agserviceuser;
GRANT GRANT ANY ROLE TO agserviceuser;
GRANT DROP USER TO agserviceuser;
GRANT SELECT on dba_roles TO agserviceuser;
GRANT SELECT ON dba_profiles TO agserviceuser;
GRANT ALTER USER TO agserviceuser;
GRANT CREATE ANY TABLE TO agserviceuser;
GRANT DROP ANY TABLE TO agserviceuser;
GRANT CREATE ANY PROCEDURE TO agserviceuser;
GRANT DROP ANY PROCEDURE TO agserviceuser;
```

Configure

You can establish a connection between Oracle Database and Oracle Access Governance by entering connection details and configuring your database environment. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field.

1. Select the **Database User Management (Oracle DB)** tile. Once selected, a value of *Database User Management (Oracle DB)* is displayed on the right hand side under **What I've selected**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the specified database.

1. In the **Easy Connect URL for Database** field, enter the connect string for the database you want to integrate with Oracle Access Governance. For use the format *host/port/database service/sid*. For use the format *jdbc:oracle:thin:@<SERVICE_NAME>?TNS_ADMIN=<WALLET-DIR>* as described in Configure Wallet for Autonomous Database Integration.
2. In the **User Name** field, enter the DB user you will use to connect to the database. This is the user you created in Create a Target System User Account for Database User Management (Oracle) Orchestrated System Operations.
3. Enter the password of the target database user in the **Password** field. Confirm the password in the **Confirm password** field.
4. In **Connection Properties** enter any connection properties in the format *prop1=val1#prop2=val2*
5. Check the right hand pane to view **What I've selected.** If you are happy with the details entered, select **Add** to create the orchestrated system.

Finish up

On the **Finish Up** step of the workflow, you are asked to download the agent you will use to interface between Oracle Access Governance and Oracle Database. Select the **Download** link to download the agent zip file to the environment in which the agent will run.

After downloading the agent, follow the instructions explained in the Agent Administration article.

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

Configure Wallet for Autonomous Database Integration

A connection to requires the client, in this case the Oracle Access Governance agent, to be configured to support SSL communication between the agent and the database service. To enable this feature, you should download the autonomous database wallet to your agent host, and then update the **Easy Connect URL for Database** field in the orchestrated system configuration. Complete the following steps to configure this feature:

1. Create a Database User Management (Oracle) orchestrated system and configure the agent.
2. Download the autonomous database wallet using the instructions in Download Client Credentials (Wallets).
3. Create a wallet directory inside the installed agent folder, `<PERSISTENT_VOLUME_LOCATION><WALLET-DIR>`. For example:

   ```
   mkdir /myagent/install/db-wallet
   ```

4. Copy the zipfile containing the wallet you downloaded in Step 2, to the `<PERSISTENT_VOLUME_LOCATION><WALLET-DIR>` and unzip using the command:

   ```
   cp -rf Wallet_<DATABASENAME>.zip <PERSISTENT_VOLUME_LOCATION><WALLET-DIR>
   ```

5. The unzipped wallet file will contain the `tnsnames.ora` file, which contains the service names available for the . Choose from one of the following depending on your workload:
   - *databasename*_tpurgent
   - *databasename*_tp
   - *databasename*_high
   - *databasename*_medium
   - *databasename*_low

   For further details on service names see Database Service Names for Autonomous Transaction Processing and Autonomous JSON Database.
6. Edit the integration settings for your orchestrated system by following the instructions in Configure settings for an Orchestrated System. Update the **Easy Connect URL for Database** field with the connect string for your database, based on the service name you selected in the previous step. The connect string should take the following format:

   ```
   jdbc:oracle:thin:@<SERVICE_NAME>?TNS_ADMIN=<WALLET-DIR>
   ```

   For example:

   ```
   jdbc:oracle:thin:@MYAUTDB_TP?TNS_ADMIN=/agent/install
   ```

Database Application Tables

Overview: Integrate Oracle Access Governance with Database Application Tables (Oracle)

Oracle Access Governance can be integrated with Database Application Tables (Oracle), enabling identity orchestration, including on-boarding of identity (user) data, and provisioning of accounts.

Database Application Tables (Oracle) can be defined as database-driven custom applications. In terms of identity administration these applications are characterized by the following:

- Applications have no APIs for identity administration
- Each application can have a different set of schemas for identity, account, and permission data
- The application schema is not known to Oracle Access Governance prior to configuring and running the integration
- There is no direct mapping between the application database tables and any Oracle Access Governance entities
- There is no common pattern to how identity data is organized within the application schema. Identity data may be in a single table, or spread across many tables

The Database Application Tables (Oracle) integration enables the exchange of user data between a customer database and Oracle Access Governance.

The Database Application Tables (Oracle) integration supports the following elements:

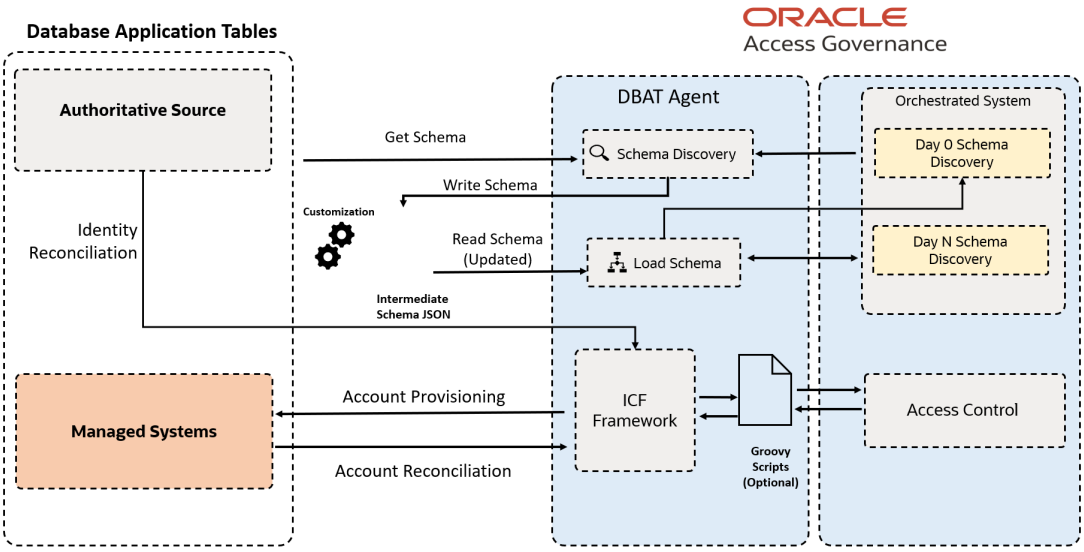
- Database Application Tables (Oracle) as an authoritative (trusted) source of identity information allowing for reconciliation of identities created or modified in database tables.
- Database Application Tables (Oracle) as a Managed System enabling provisioning of application accounts in database tables.

Database Application Tables (Oracle) Integration Architecture Overview

Integration with Database Application Tables (Oracle) allows you to retrieve identity data from a system, transport it to Oracle Access Governance, and ingest. Once a system is connected, you can perform provisioning and remediation tasks and operations, such as create, reconcile,

update, delete, disable/enable, add/delete permission, and add/delete group, which are then reflected in the managed system.

Figure 5-1 DBAT Architecture



Database Application Tables (Oracle) integration is implemented using an Agent-based connection type. This means that a direct connection is not available, so an indirect connection is made between Oracle Access Governance and the required database instance using the Access Governance Agent. The Database Application Tables (Oracle) integration supports the following modes:

- If you select the Authoritative Source configuration mode when you setup a Database Application Tables (Oracle) Orchestrated System, then Oracle Access Governance will retrieve identity data from the database instance and treat it as an authoritative (trusted) source of identity information.
- If you select the Managed Systems configuration mode, then Oracle Access Governance will allow you to manage user accounts in the target database. This enables the provisioning of new accounts in customer database instances from Oracle Access Governance.

In order to identify the relevant schema for your integration, the Database Application Tables (Oracle) integration offers automatic schema discovery. This is the process of identifying the underlying database schema that contains your user data. Support is provided for Day-0 limited schema, and subsequent modifications in the user schema are accommodated by Day-N support, which allows you to update the schema with any changes that have been made to the database tables, and apply them to your discovered schema

Details of the relevant user database tables are provided during agent creation. Details of the schema are stored in a schema file, which is located with the agent. Details of the schema can be updated by directly editing this file as required.

Connection to the database holding your user tables is made through an JDBC database connection. This allows the agent to:

- Discover the schema for your database-driven application

- Perform dataloads
- Provision accounts

A full load of relevant identity and account data is made into Oracle Access Governance each time the load is executed. If this is the first time that the load is made, then relevant identity and account structures are created in Oracle Access Governance as appropriate. On subsequent data load runs, all data is loaded to Oracle Access Governance and the ingestion process updates any changes since the last data load in the appropriate identity and account artefacts.

Once your orchestrated system is configured, you can provision accounts using Oracle Access Governance's provisioning engine which will take any request for accounts or permissions and pass it through the agent and onwards to the target database. Provisioning supports create, update, and revoke operations.

In the event that your schema file is lost or corrupted in any way, the Database Application Tables (Oracle) integration provides schema file recovery. This is useful in scenarios such as an agent crash or loss of the schema file.

Database Application Tables (Oracle) Integration Functional Overview

Database Application Tables (Oracle) integration supports usecases including configuration of the Orchestrated System, dataload, account creation and revocation, change password, and assign and remove roles.

Supported Integration Functions

Integration of Database Application Tables (Oracle) with Oracle Access Governance supports the following functions:

- Configure Orchestrated System
- Load Data
- Create Account
- Assign Permissions
- Remove Permissions
- Change Password
- Revoke Account

For full details of functions supported, refer to Oracle Access Governance Integration Functional Overview

An Example Account Lifecycle

Let's look at an example. You have created a new orchestrated system which is connected to the **MyDBAT** database instance which contains user data for your organization. The orchestrated system is configured for Authoritative Source and Managed System modes. On the first dataload, identity and account data is loaded into Oracle Access Governance. At this time the following details are created in Oracle Access Governance:

- An Oracle Access Governance identity is created, say*MyAGIdentity*, comprising authoritative data such as name, email, and location.
- An account is created in Oracle Access Governance for existing Database Application Tables (Oracle) roles, say *DBATRole_Composer*.

We now have the following:

- **MyAGIdentity**

  - MyDBATAccount
    * DBATRole_Composer

After some time *MyAGIdentity* moves into a new position within their organization requiring the developer role. An access bundle *DBATBundle_Developer* is created in Oracle Access Governance which contains the development permissions required. This access bundle can be assigned as a result of a policy, role or request. Let's say the user requests the access bundle using the **Request a new access** option. On approval, the request triggers a provisioning operation which applies the changes to **MyDBAT**, assigning the Database Application Tables (Oracle) roles corresponding to the permissions contained in *DBATBundle_Developer* access bundle.

We now have the following:

- **MyAGIdentity**
  - MyDBATAccount
    * DBATRole_Composer
    * DBATBundle_Developer

Additional accounts may be mapped to the *MyAGIdentity* identity over time from other managed systems giving us a profile like this:

- **MyAGIdentity**
  - MyDBATAccount
  - MyOracleDBAccount
  - MyMSTeamsAccount

MyAGIdentity is then required to change their password. Using the **My Access** functionality in Oracle Access Governance Console, they change their password, which propagates the change to **MyDBAT** using Oracle Access Governance provisioning.

MyAGIdentity then moves into a role which means they no longer require an account on **MyDBAT**. In this case a revoke account provisioning task can be generated by revoking the identity's account as part of an access review. Alternatively, their association with customer database roles can be removed by removing the identity from the relevant Oracle Access Governance role or policy. In either case, this will result in a provisioning task which will revoke the account from the customer database, together with any related roles. The profile would now resemble:

- **MyAGIdentity**
  - MyOracleDBAccount
  - MyMSTeamsAccount

If the Database Application Tables (Oracle) Orchestrated System is configured in *Authoritative Source* mode, and you make an identity inactive, then the the *MyAGIdentity* identity is effectively disabled. In this case a provisioning task will be generated and applied to the Managed System.

We now have the following:

- **MyAGIdentity (Disabled)**

Prerequisites

Before you install and configure a Database Application Tables (Oracle) Orchestrated System, you should consider the following prerequisites and tasks.

1. Your Database Application Tables (Oracle) system is certified with Oracle Access Governance. Refer to Database Application Tables (Oracle) Components Certified for Integration with Oracle Access Governance for details of the versions supported.

Configure

You can establish a connection between customer databases and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field.

1. Select **Database Application Table (Oracle DB)**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the details required to allow Oracle Access Governance to connect to your customer database.

Table 5-14 Integration settings

Parameter Name	Mandatory?	Description
Easy Connect URL for Oracle database	Yes	URL of the server hosting the customer database system you want to integrate with. For use the format *host/port/database service/sid*. For use the format *jdbc:oracle:thin:@<SERVICE_NAME>?TNS_ADMIN=<WALLET-DIR>* as described in Configure Wallet for Autonomous Database Integration.
User name	Yes	The username required to connect to the user database system to perform data reconciliation and provisioning.
Password/Confirm password	Yes	The password that authenticates the user you are connecting to the user database system with.
User account table name	Yes	The name of the database table containing your user accounts. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_PERSON else you will see errors. User name is passed as a separate parameter as detailed in this table.

Table 5-14 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Permission tables		Add the names of your permission tables in a comma-separated list. This parameter only applies if your orchestrated system is configured in managed system mode. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_PERMISSION else you will see errors. User name is passed as a separate parameter as detailed in this table.
Account permission tables		If you have account data resident in parent and child tables, then provide a comma-separated list of the child tables names. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_ACCOUNTPERMISSION else you will see errors. User name is passed as a separate parameter as detailed in this table.

Table 5-14 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Lookup tables		Comma-separated list of lookup tables for attributes such as country. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_LOOKUP else you will see errors. User name is passed as a separate parameter as detailed in this table.
Key column mappings	Yes	Comma-separated list of key column mappings. These mappings should be entered in the format `Table:KeyColumn`. **Note:** This parameter is applicable for ACCOUNT, ENTITLEMENT, and LOOKUP tables only.
Name column mappings	Yes	Comma-separated list of name column mappings. These mappings should be entered in the format `Table:NameColumn`. **Note:** This parameter is applicable for ACCOUNT, ENTITLEMENT, and LOOKUP tables only.
User account table password column mapping		Password column mapping for user account table in the format `Table:PasswordColumn`.

Table 5-14 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
User account table status column mapping	Yes	Status column mapping for the user account table in the format `Table:StatusColumn`. The status column holds the status of a user record. In case of special values, please configure the enable/ disabled value.
User account enabled status value		This value will be used as the enable value if the status column is configured, and it is a String type. If no value is provided for this parameter, then it defaults to 'ACTIVE'.
User account disabled status value		This value will be used as the disable value if the status column is configured, and it is a String type. If no value is provided for this parameter, then it defaults to 'INACTIVE'.
Date format		Format for date data that is being converted to strings. If you want to handle date data as a date editor, then do not enter any value for this parameter. If you want to handle date data as text, then you must enter the date format. Specifying a value for this parameter invalidates the allNative parameter.
Timestamp format		Format for timestamp data that is being converted to strings. Specifying this property invalidates the nativeTimestamps and allNative properties
User account filter condition		A WHERE clause which defines the subset of user account records that you want to bring from your customer database into Oracle Access Governance.
Create script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the create user account provisioning operation. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/create_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy

Table 5-14 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Update script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the update user account provisioning operation. This script is called when you update the account attribute form, enable or disable the user account. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/update_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy
Delete script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the delete user account provisioning operation. This script is called when you revoke or delete an account. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/delete_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy
Dataload script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for reconciliation. The connector delegates the data load operation to the Groovy script, which is responsible for passing the information (connector object) to the callback handler. This script is called while performing an account search (operations such as full data load). You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/full_data_load.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy

Table 5-14 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Add relationship data script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the add multivalued attribute (including permissions for account) provisioning operation. This script is called when you add multivalued child attributes. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/add_mulval_attr.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy
Remove relationship data script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the remove multivalued attribute (including permissions for account) provisioning operation. This script is called while removing multivalued child attributes. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/remove_mulval_attr.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy

1. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up** where you are prompted to download the agent for your Orchestrated System. Once you have downloaded the agent, you can install and configure the agent in your environment using the instructions in Manage Oracle Access Governance Agent for Indirect Integrations.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

Update Intermediate Schema JSON File

When you have completed installation of your agent, an intermediate schema JSON file, `schema.json` is created on the agent host. This file maps the tables in the integrated database with the schema which is represented on Oracle Access Governance. The initial schema JSON file is created with basic attributes enabled for data load, UID, NAME, STATUS and PASSWORD (if configured by user). The full data load operation can execute with this initial schema JSON file, loading data for only these basic attributes. You can then further modify the schema JSON file to include more attributes for the next data load operations.

> **Note:**
>
> Ensure that you have granted read/write permissions on the schema JSON file for the operating system user that will be running the agent.

For full details on the structure and options available when editing the `schema.json`, refer to Schema JSON File Reference.

Fetch Latest Custom Attributes

You should perform a schema discovery operation which will fetch the latest custom attribute information. For details on how to perform this task, see Fetch Latest Custom Attributes.

Configuring SSL/TLS Communication in Oracle Database

To secure communication between your Oracle Access Governance agent and the Oracle database, you can configure Secure Sockets Layer/Transport Layer Security (SSL/TLS). To do this, ensure that you have completed the following steps:

1. **Configure Data Encryption and Integrity in Oracle Database**
   See Configuring Transport Layer Security Authentication for information about configuring data encryption and integrity.
2. To configure your Oracle Access Governance agent to use SSL/TLS when communicating with the database, perform the following steps:
   a. Export the certificate on the Oracle Database host computer.
   b. Copy the database certificate to your Oracle Access Governance agent host.
   c. Import the database certificate into the Java truststore of the agent using the command:

      ```
      <%JAVA_HOME%>/bin/keytool -import -alias database-cert -file <AD-cert-file> -keystore <agent-install-dir>/cacerts
      ```

   d. Update the agent `config.properties` file to include the following:

      ```
      JAVA_OPTS=-Djavax.net.ssl.trustStore=/app/cacerts-
      Djavax.net.ssl.trustStorePassword=changeit
      ```

Configure Wallet for Autonomous Database Integration

A connection to requires the client, in this case the Oracle Access Governance agent, to be configured to support SSL communication between the agent and the database service. To enable this feature, you should download the autonomous database wallet to your agent host, and then update the **Easy Connect URL for Database** field in the orchestrated system configuration. Complete the following steps to configure this feature:

1. Create a Database User Management (Oracle) orchestrated system and configure the agent.
2. Download the autonomous database wallet using the instructions in Download Client Credentials (Wallets).
3. Create a wallet directory on the agent host. For example:

   ```
   mkdir /app/db-wallet
   ```

4. Copy the zipfile containing the wallet you downloaded in Step 2, to the wallet folder, and unzip using the command:

   ```
   cp -rf Wallet_<DATABASENAME>.zip /app/db-wallet
   ```

5. The unzipped wallet file will contain the `tnsnames.ora` file, which contains the service names available for the . Choose from one of the following depending on your workload:
   - *databasename*_tpurgent
   - *databasename*_tp
   - *databasename*_high
   - *databasename*_medium
   - *databasename*_low

   For further details on service names see Database Service Names for Autonomous Transaction Processing and Autonomous JSON Database.
6. Edit the integration settings for your orchestrated system by following the instructions in Configure settings for an Orchestrated System. Update the **Easy Connect URL for Database** field with the connect string for your database, based on the service name you selected in the previous step. The connect string should take the following format:

   ```
   jdbc:oracle:thin:@<SERVICE_NAME>?TNS_ADMIN=<WALLET-DIR>
   ```

   For example:

   ```
   jdbc:oracle:thin:@MYAUTDB_TP?TNS_ADMIN=/app/db-wallet
   ```

Prerequisites

Before you install and configure a Orchestrated System, you should consider the following prerequisites and tasks.

1. Your system is certified with Oracle Access Governance. Refer to Database Application Tables (Oracle) Components Certified for Integration with Oracle Access Governance for details of the versions supported.

Configure

You can establish a connection between customer databases and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field.

1. Select **Database Application Table (MSSQL DB)**.
2. Click **Next**.

Add details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the details required to allow Oracle Access Governance to connect to your customer database.

Table 5-15 Integration settings

Parameter Name	Mandatory?	Description
Easy connect URL for Microsoft SQL Server database	Yes	URL of the server hosting the customer database system you want to integrate with. Use the format *host/port/database/encrypt/trustServerCertificate*, for example *jdbc:sqlserver://[host]:[port];[databaseName];[encrypt];[trustServerCertificate]*. For further details refer to the MSSQL JDBC documentation for your version.
User name	Yes	The username required to connect to the customer database system to perform data reconciliation and provisioning.
Password?/Confirm password	Yes	The password that authenticates the user you are connecting to the customer database system with.
Database name	Yes	The customer database to which you need to connect (such as master database).
Custom jar details	Yes	MSSQL Server database driver jar. Please refer to the MSSQL JDBC documentation for your version for details. The jar name and checksum should be in the format of `<jarName>::<jarChecksum>`.Calculate the checksum using SHA-512. Details of how this is used by the agent can be found in Custom Jar Support.

Table 5-15 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
User account table name	Yes	The name of the table containing your user accounts. **Note:** For a key column which is non auto increment, Create Account provisioning will be supported only with custom script. For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MSSQL) Using Groovy. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_PERSON else you will see errors. User name is passed as a separate parameter as detailed in this table.

Table 5-15 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Permission tables		Add the names of your permission tables in a comma-separated list. This parameter only applies if your orchestrated system is configured in managed system mode. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_PERMISSION else you will see errors. User name is passed as a separate parameter as detailed in this table.
Account permission tables		If you have account data resident in parent and child tables, then provide a comma-separated list of the child tables names. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_ACCOUNTPERMISSION else you will see errors. User name is passed as a separate parameter as detailed in this table.

Table 5-15 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Lookup tables		Comma-separated list of lookup tables for attributes such as country. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_LOOKUP else you will see errors. User name is passed as a separate parameter as detailed in this table.
Key column mappings	Yes	Comma-separated list of key column mappings. These mappings should be entered in the format `Table:KeyColumn`. **Note:** This parameter is applicable for ACCOUNT, ENTITLEMENT, and LOOKUP tables only.
Name column mappings	Yes	Comma-separated list of name column mappings. These mappings should be entered in the format `Table:NameColumn`. **Note:** This parameter is applicable for ACCOUNT, ENTITLEMENT, and LOOKUP tables only.
User account table password column mapping		Password column mapping for user account table in the format `Table:PasswordColumn`.

Table 5-15 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
User account table status column mapping	Yes	Status column mapping for the user account table in the format `Table:StatusColumn`. The status column holds the status of a user record. In case of special values, please configure the enable/ disabled value.
User account enabled status value		This value will be used as the enable value if the status column is configured, and it is a String type. If no value is provided for this parameter, then it defaults to 'ACTIVE'.
User account disabled status value		This value will be used as the disable value if the status column is configured, and it is a String type. If no value is provided for this parameter, then it defaults to 'INACTIVE'.
User account filter condition		A WHERE clause which defines the subset of user account records that you want to bring from your customer database into Oracle Access Governance. For example: COUNTRY in ('IN','US').
Create script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the create user account provisioning operation. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/create_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MSSQL) Using Groovy.
Update script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the update user account provisioning operation. This script is called when you update the account attribute form, enable or disable the user account. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/update_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MSSQL) Using Groovy.

Table 5-15 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Delete script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the delete user account provisioning operation. This script is called when you revoke or delete an account. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/delete_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MSSQL) Using Groovy .
Dataload script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for reconciliation. The connector delegates the data load operation to the Groovy script, which is responsible for passing the information (connector object) to the callback handler. This script is called while performing an account search (operations such as full data load). You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/full_data_load.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MSSQL) Using Groovy.
Add relationship data script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the add multivalued attribute (including permissions for account) provisioning operation. This script is called when you add multivalued child attributes. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/add_mulval_attr.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MSSQL) Using Groovy.

Table 5-15 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Remove relationship data script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the remove multivalued attribute (including permissions for account) provisioning operation. This script is called while removing multivalued child attributes. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/remove_mulval_attr.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MSSQL) Using Groovy.

1. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up** where you are prompted to download the agent for your Orchestrated System. Once you have downloaded the agent, you can install and configure the agent in your environment using the instructions in Manage Oracle Access Governance Agent for Indirect Integrations.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

Update Intermediate Schema JSON File

When you have completed installation of your agent, an intermediate schema JSON file, `schema.json` is created on the agent host. This file maps the tables in the integrated database with the schema which is represented on Oracle Access Governance. The initial schema JSON file is created with basic attributes enabled for data load, UID, NAME, STATUS and PASSWORD (if configured by user). The full data load operation can execute with this initial schema JSON file, loading data for only these basic attributes. You can then further modify the schema JSON file to include more attributes for the next data load operations.

> **Note:**
>
> Ensure that you have granted read/write permissions on the schema JSON file for the operating system user that will be running the agent.

For full details on the structure and options available when editing the `schema.json`, refer to Schema JSON File Reference.

Fetch Latest Custom Attributes

You should perform a schema discovery operation which will fetch the latest custom attribute information. For details on how to perform this task, see Fetch Latest Custom Attributes.

Prerequisites

Before you install and configure a Database Application Tables (Oracle) Orchestrated System, you should consider the following prerequisites and tasks.

1. Your Database Application Tables (Oracle) system is certified with Oracle Access Governance. Refer to Database Application Tables (Oracle) Components Certified for Integration with Oracle Access Governance for details of the versions supported.

Configure

You can establish a connection between customer databases and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field.

1. Select **Database Application Table (MySQL)**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

The default value in each case is *Selected*.

4. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the details required to allow Oracle Access Governance to connect to your customer database.

Table 5-16 Integration settings

Parameter Name	Mandatory?	Description
What is the easy connect URL for MySQL database?	Yes	URL of the server hosting the customer database system you want to integrate with. Use the format *host/port/database service/sid*, for example *jdbc:mysql:@[host]:[port]:[sid]*. For further details refer to the MySQL JDBC documentation for your version.
What is the username for authentication?	Yes	The username required to connect to the customer database system to perform data reconciliation and provisioning.
What is the password?/Confirm password	Yes	The password that authenticates the user you are connecting to the customer database system with.
Custom jar details	Yes	MySQL Server database driver jar. Please refer to the MySQL JDBC documentation for your version for details. The jar name and checksum should be in the format of `<jarName>::<jarChecksum>`.Calculate the checksum using SHA-512. Details of how this is used by the agent can be found in Custom Jar Support.

Table 5-16 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
User account table name	Yes	The name of the table containing your user accounts. **Note:** For a key column which is non auto increment, Create Account provisioning will be supported only with custom script. For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MySQL) Using Groovy. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_PERSON else you will see errors. User name is passed as a separate parameter as detailed in this table.

Table 5-16 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Permission tables		Add the names of your permission tables in a comma-separated list. This parameter only applies if your orchestrated system is configured in managed system mode. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_PERMISSION else you will see errors. User name is passed as a separate parameter as detailed in this table.
Account permission tables		If you have account data resident in parent and child tables, then provide a comma-separated list of the child tables names. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_ACCOUNTPERMISSION else you will see errors. User name is passed as a separate parameter as detailed in this table.

Table 5-16 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Lookup tables		Comma-separated list of lookup tables for attributes such as country. **Note:** Do not include the user name of the table owner in the table name e.g. MYUSER.MYDBAT_LOOKUP else you will see errors. User name is passed as a separate parameter as detailed in this table.
Key column mappings	Yes	Comma-separated list of key column mappings. These mappings should be entered in the format `Table:KeyColumn`. **Note:** This parameter is applicable for ACCOUNT, ENTITLEMENT, and LOOKUP tables only.
Name column mappings	Yes	Comma-separated list of name column mappings. These mappings should be entered in the format `Table:NameColumn`. **Note:** This parameter is applicable for ACCOUNT, ENTITLEMENT, and LOOKUP tables only.
User account table password column mapping		Password column mapping for user account table in the format `Table:PasswordColumn`.

Table 5-16 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
User account table status column mapping	Yes	Status column mapping for the user account table in the format `Table:StatusColumn`. The status column holds the status of a user record. In case of special values, please configure the enable/ disabled value.
User account enabled status value		This value will be used as the enable value if the status column is configured, and it is a String type. If no value is provided for this parameter, then it defaults to 'ACTIVE'.
User account disabled status value		This value will be used as the disable value if the status column is configured, and it is a String type. If no value is provided for this parameter, then it defaults to 'INACTIVE'.
Date format		Format for date data that is being converted to strings. If you want to handle date data as a date editor, then do not enter any value for this parameter. If you want to handle date data as text, then you must enter the date format. Specifying a value for this parameter invalidates the allNative parameter.
Timestamp format		Format for timestamp data that is being converted to strings. Specifying this property invalidates the nativeTimestamps and allNative properties
User account filter condition		A WHERE clause which defines the subset of user account records that you want to bring from your customer database into Oracle Access Governance.
Create script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the create user account provisioning operation. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/create_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MySQL) Using Groovy.

Table 5-16 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Update script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the update user account provisioning operation. This script is called when you update the account attribute form, enable or disable the user account. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/update_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MySQL) Using Groovy.
Delete script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the delete user account provisioning operation. This script is called when you revoke or delete an account. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/delete_user.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MySQL) Using Groovy.
Dataload script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for reconciliation. The connector delegates the data load operation to the Groovy script, which is responsible for passing the information (connector object) to the callback handler. This script is called while performing an account search (operations such as full data load). You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/full_data_load.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MySQL) Using Groovy.

Table 5-16 (Cont.) Integration settings

Parameter Name	Mandatory?	Description
Add relationship data script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the add multivalued attribute (including permissions for account) provisioning operation. This script is called when you add multivalued child attributes. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/add_mulval_attr.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MySQL) Using Groovy.
Remove relationship data script		Custom script to use custom stored procedures or SQL statements rather than the default SQL statements for performing provisioning operations. Enter the file URL of the Groovy script created for the remove multivalued attribute (including permissions for account) provisioning operation. This script is called while removing multivalued child attributes. You must enter the file URL in the following format: `/directoryName/fileName`. Sample value: `/app/scripts/remove_mulval_attr.groovy` For further details on scripting with the Database Application Tables (Oracle) integration, see Develop Custom Scripts for Database Application Tables (MySQL) Using Groovy.
Connection properties		Connection properties that will be used to configure a secure connection. They should be key value pairs in the following format: key1=val1#key2=val2.

1. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up** where you are prompted to download the agent for your Orchestrated System. Once you have downloaded the agent, you can install and configure the agent in your environment using the instructions in Manage Oracle Access Governance Agent for Indirect Integrations.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

Update Intermediate Schema JSON File

When you have completed installation of your agent, an intermediate schema JSON file, `schema.json` is created on the agent host. This file maps the tables in the integrated database with the schema which is represented on Oracle Access Governance. The initial schema JSON file is created with basic attributes enabled for data load, UID, NAME, STATUS and PASSWORD (if configured by user). The full data load operation can execute with this initial schema JSON file, loading data for only these basic attributes. You can then further modify the schema JSON file to include more attributes for the next data load operations.

> **Note:**
>
> Ensure that you have granted read/write permissions on the schema JSON file for the operating system user that will be running the agent.

For full details on the structure and options available when editing the `schema.json`, refer to Schema JSON File Reference.

Fetch Latest Custom Attributes

You should perform a schema discovery operation which will fetch the latest custom attribute information. For details on how to perform this task, see Fetch Latest Custom Attributes.

Database Application Tables (Oracle) Components Certified for Integration with Oracle Access Governance

The Database Application Tables (Oracle) components that you can integrate with are listed below.

Certified Components

Table 5-17 Certified Components

Component Type	Component
Oracle Specific Requirements	
System	The target system can be database tables from any one of the following RDBMSs: • Oracle Autonomous Database • Oracle Database 23ai, 19c, 18c or 12c as a single database, pluggable database (PDB), or Oracle RAC implementation • Oracle Database 10g and 11g as either a single database or Oracle RAC implementation
JDK	JDK 1.6 or later
Microsoft SQL Server Specific Requirements	
System	Microsoft SQL Server 2016, 2017

Table 5-17 (Cont.) Certified Components

Component Type	Component
JDBC Drivers	For Microsoft SQL Server 2014: sqljdbc4 version 4.0
Microsoft MySQL Specific Requirements	
System	MySQL 5.x, MySQL 8.x
JDBC Drivers	mysql-connector-java-5.1.12-bin
General Requirements	
Format in which user data is stored in the system	You can use a Database Application Tables connector only if user data is stored in the target system in any one of the following formats: • All user data is in a single table or view. • User data is spread across one parent table and one or more child tables. This target system can be configured only as a managed system, and not as an authoritative source. • All user data is in a single updatable view (that is based on one or more tables). • User data is spread across one updatable view (that is based on one or more tables) and one or more child views (that are based on one or more tables). This type of system can be configured only as a managed system, and not as an authoritative source with this connector. In other words, an authoritative source cannot store child data.
Other requirements of the system	The system must meet the following requirement: • If parent and child tables are not joined by a foreign key (for example, if you are using views), then the names of the foreign key columns in both tables must be the same. • The primary key for any tables used in the target system should be provided by a single column for identity, account, entitlement, and lookup tables. Composite primary keys are only supported for tables linking entitlement and identity/account tables.

Supported Configuration Modes for Database Application Tables (Oracle) Integrations

Oracle Access Governance integrations can be setup in different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Supported Modes

Database Application Tables (Oracle) Orchestrated System supports the following modes:

- **Authoritative Source**
  You can use Database Application Tables (Oracle) as an authoritative (trusted) source of identity information for Oracle Access Governance where:
  - All user data is in a single table or view OR
  - All user data is in a single updatable view (that is based on one or more tables)

  .

- **Managed System**
  You can manage Database Application Tables (Oracle) permissions where:
  - All user data is in a single table or view OR
  - All user data is in a single updatable view (that is based on one or more tables) OR
  - User data is spread across one parent table and one or more child tables. OR
  - User data is spread across one updatable view (that is based on one or more tables) and one or more child views (that are based on one or more tables).

Configure A Minimum Privileged Service User

Configure A Minimum Privileged Service User For

To enable a secure connection between Oracle Access Governance and the database, you can create a service user with the minimum privileges required to configure the integration.

Permissions Required for Authoritative Source Mode

If you configure your orchestrated system in authoritative mode, you need to grant read permissions to your service user on the table containing your identities, so that they can be loaded into Oracle Access Governance. The minimum set of permissions required in this case is `SELECT` permission on the table containing identity or person information.

An example might be:

```
GRANT SELECT ON MYDBAT_PERSON TO SERVICEUSER;
```

where:

- `MYDBAT_PERSON`: Is the table in your database containing identity information.
- `SERVICEUSER`: Is the service account user.

Permissions Required for Managed System Mode

If you configure your orchestrated system in managed system mode, you need to grant read and write permissions for account tables and permissions tables, to allow for reconciliation, create, update, and delete of accounts and account permissions. The minimum set of permissions required in this case is `SELECT`, `INSERT`, `UPDATE`, and `DELETE` permissions on the account and account permission tables.

An example might be:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_ROLES TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_GROUPS TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON_ROLE TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON_GROUP TO SERVICEUSER;
```

where

- `MYDBAT_PERSON`: Is the table in your database containing account information.
- `MYDBAT_ROLES`: Is the table in your database containing role information.
- `MYDBAT_GROUPS`: Is the table in your database containing group information.
- `MYDBAT_PERSON_ROLE`: Is the table in your database containing people role information.
- `MYDBAT_PERSON_GROUP`: Is the table in your database containing group role information.
- `SERVICEUSER`: Is the service account user.

Permissions Required for Custom Scripts

If you want to develop custom scripts for operations on the integration as described in Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy, you will need to add permissions to any stored procedures or other database objects referenced in your custom scripts. If you create the stored procedures using the service user then you should not require any permissions. If stored procedures or other database objects are created in another user, then you should grant permissions as appropriate.

An example might be:

```
GRANT EXECUTE ON MYDBAT_CUSTOMDEV.GET_USERROLE TO SERVICEUSER;
```

where:

- `MYDBAT_CUSTOMDEV`: Is the user you used to create the stored procedure.
- `GET_USERROLE` is the stored procedure name.
- `SERVICEUSER`: Is the service account user.

Configure A Minimum Privileged Service User For

To enable a secure connection between Oracle Access Governance and the database, you can create a service user with the minimum privileges required to configure the integration.

Permissions Required for Authoritative Source Mode

If you configure your orchestrated system in authoritative mode, you need to grant read permissions to your service user on the table containing your identities, so that they can be

loaded into Oracle Access Governance. The minimum set of permissions required in this case is `SELECT` permission on the table containing identity or person information.

An example might be:

```
GRANT SELECT ON MYDBAT_PERSON TO SERVICEUSER;
```

where:

- `MYDBAT_PERSON`: Is the table in your database containing identity information.
- `SERVICEUSER`: Is the service account user.

Permissions Required for Managed System Mode

If you configure your orchestrated system in managed system mode, you need to grant read and write permissions for account tables and permissions tables, to allow for reconciliation, create, update, and delete of accounts and account permissions. The minimum set of permissions required in this case is `SELECT`, `INSERT`, `UPDATE`, and `DELETE` permissions on the account and account permission tables.

An example might be:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_ROLES TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_GROUPS TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON_ROLE TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON_GROUP TO SERVICEUSER;
```

where

- `MYDBAT_PERSON`: Is the table in your database containing account information.
- `MYDBAT_ROLES`: Is the table in your database containing role information.
- `MYDBAT_GROUPS`: Is the table in your database containing group information.
- `MYDBAT_PERSON_ROLE`: Is the table in your database containing people role information.
- `MYDBAT_PERSON_GROUP`: Is the table in your database containing group role information.
- `SERVICEUSER`: Is the service account user.

Permissions Required for Custom Scripts

If you want to develop custom scripts for operations on the integration as described in Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy, you will need to add permissions to any stored procedures or other database objects referenced in your custom scripts. If you create the stored procedures using the service user then you should not require any permissions. If stored procedures or other database objects are created in another user, then you should grant permissions as appropriate.

An example might be:

```
GRANT EXECUTE ON OBJECT::dbo.GET_USERROLE TO SERVICEUSER;
```

where:

- `OBJECT::dbo`: Is the Microsoft MSSQL database object.
- `GET_USERROLE` is the stored procedure name.
- `SERVICEUSER`: Is the service account user.

Configure A Minimum Privileged Service User For

To enable a secure connection between Oracle Access Governance and the database, you can create a service user with the minimum privileges required to configure the integration.

Permissions Required for Authoritative Source Mode

If you configure your orchestrated system in authoritative mode, you need to grant read permissions to your service user on the table containing your identities, so that they can be loaded into Oracle Access Governance. The minimum set of permissions required in this case is `SELECT` permission on the table containing identity or person information.

An example might be:

```
GRANT SELECT ON MYDBAT_PERSON TO SERVICEUSER;
```

where:

- `MYDBAT_PERSON`: Is the table in your database containing identity information.
- `SERVICEUSER`: Is the service account user.

Permissions Required for Managed System Mode

If you configure your orchestrated system in managed system mode, you need to grant read and write permissions for account tables and permissions tables, to allow for reconciliation, create, update, and delete of accounts and account permissions. The minimum set of permissions required in this case is `SELECT`, `INSERT`, `UPDATE`, and `DELETE` permissions on the account and account permission tables.

An example might be:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_ROLES TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_GROUPS TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON_ROLE TO SERVICEUSER;

GRANT SELECT, INSERT, UPDATE, DELETE ON MYDBAT_PERSON_GROUP TO SERVICEUSER;
```

where

- `MYDBAT_PERSON`: Is the table in your database containing account information.
- `MYDBAT_ROLES`: Is the table in your database containing role information.
- `MYDBAT_GROUPS`: Is the table in your database containing group information.
- `MYDBAT_PERSON_ROLE`: Is the table in your database containing people role information.
- `MYDBAT_PERSON_GROUP`: Is the table in your database containing group role information.
- `SERVICEUSER`: Is the service account user.

Permissions Required for Custom Scripts

If you want to develop custom scripts for operations on the integration as described in Develop Custom Scripts for Database Application Tables (Oracle) Using Groovy, you will need to add permissions to any stored procedures or other database objects referenced in your custom scripts. If you create the stored procedures using the service user then you should not require any permissions. If stored procedures or other database objects are created in another user, then you should grant permissions as appropriate.

An example might be:

```
GRANT EXECUTE ON MYSQLDB.GET_USERROLE TO 'SERVICEUSER';
```

where:

- `MYSQLDB`: Is the MySQL database in which you created the stored procedure.
- `GET_USERROLE` Is the stored procedure name.
- `SERVICEUSER`: Is the service account user.

Supported Operations When Provisioning To Database Application Tables (Oracle)

When you provision an account from Oracle Access Governance to Database Application Tables (Oracle) certain operations are supported.

The Database Application Tables (Oracle) Orchestrated System supports the following account operations when provisioning a user:

- Create account
- Enable account
- Disable account
- Revoke account
- Assign permission
- Remove permission

Supported Data Types

The data types supported for reconciliation and provisioning operations are listed in the following section:

For Oracle Database

The data types supported for reconciliation and provisioning operations for an Oracle Database orchestrated system are as listed below:

- VARCHAR2
- CHAR
- NUMBER
- NUMERIC
- INTEGER
- INT
- SMALLINT
- DOUBLE
- FLOAT
- DECIMAL
- DEC
- REAL
- DATE
- TIMESTAMP

For Microsoft SQL Server

The data types supported for reconciliation and provisioning operations for a Microsoft SQL Server database orchestrated system are as listed below:

- CHAR
- VARCHAR
- SMALLINT
- INT
- BIGINT
- DECIMAL
- NUMERIC
- NVARCHAR
- FLOAT
- REAL
- SMALLDATETIME
- DATETIME

For MySQL

The data types supported for reconciliation and provisioning operations for an MySQL database orchestrated system are as listed below:

- BOOL

- SMALLINT
- MEDIUMINT
- INT
- BIGINT
- FLOAT
- DOUBLE
- DECIMAL
- CHAR
- VARCHAR
- TINYTEXT
- DATE
- DATETIME
- TIMESTAMP

Default Supported Attributes

As Database Application Tables (Oracle) integration requires schema discovery, and the discovered schema is not fixed, there are no specific default supported attributes as such. You can modify your `schema.json` to add core and custom attributes as required. As a minimum you should include the `uid` and `name` attributes required for an Oracle Access Governance identity as defined in Core Identity Attributes.

Default Matching Rules

In order to map accounts to identities in Oracle Access Governance you need to have a matching rule for each Orchestrated System.

The default matching rule for Database Application Tables (Oracle) orchestrated system is:

Table 5-18 Default Matching Rules

Mode	Default Matching Rule

Authoritative Source
Identity matching checks if incoming identities match an existing identity or are new
	Screen value:
	`Employee user name = Employee user name`
	Attribute name:
	`Identity.userName = Identity.userName`

Managed System
Account matching checks if incoming accounts match with existing identities.
	Screen value:
	`User login = Employee user name`
	Attribute name:
	`Account.name = Identity.userName`

Database Access Table Guidelines

When using database access tables with Oracle Access Governance you should consider the following guidelines in the design and structure of your tables.

General Guidelines

Any database access tables should conform to the following guidelines:

- The **Name** and **Key** columns for any entity must be configured as NOT NULL.
- For a given entity, the same database column can be configured as the **Name** and the **Key**. You can also use different columns for these if required.
- Any core attribute included in the ACCOUNT entity should comply with the following rules:
  - The column data type should be compatible with the Oracle Access Governance-side data type.
  - Any attribute that is configured as `"nature":["REQUIRED"]` in the schema JSON file should correspond to a NOT NULL database field.
- ENTITLEMENT and LOOKUP tables must have a primary key constraint on key columns.
- If related columns on ENTITLEMENT and LOOKUP tables match then there is no need to define a foreign key relationship between your ACCOUNT/TARGETACCOUNT tables and the ENTITLEMENT and LOOKUP tables. See the following section for more details.

Table Relationships

User tables which map to ACCOUNT/TARGETACCOUNT entities may have a relationship with ENTITLEMENT and LOOKUP tables. When the Database Application Tables (Oracle) connector performs schema discovery, it will initially look for foreign keys (FK) defined as constraints in your database tables which define the relationship between your user table and your ENTITLEMENT or LOOKUP tables. This would look something like the example that follows for the relationship between a user table and a lookup table for the user's country:

```
CREATE TABLE MYDBAT_PERSON
  (USERID VARCHAR2 NOT NULL ENABLE,
   USERNAME VARCHAR2 NOT NULL ENABLE,
   FIRSTNAME VARCHAR2,
   LASTNAME VARCHAR2,
   EMAIL VARCHAR2 NOT NULL ENABLE,
   HOMECOUNTRY VARCHAR2,
   FOREIGN KEY (HOMECOUNTRY) REFERENCES MYDBAT_COUNTRY(COUNTRYCODE));

CREATE TABLE MYDBAT_COUNTRY
  (COUNTRYCODE VARCHAR2 NOT NULL ENABLE,
   COUNTRYNAME VARCHAR2 NOT NULL ENABLE,
   CONSTRAINT MYDBAT_COUNTRY_PK PRIMARY KEY (COUNTRYCODE)
```

If no foreign key is defined then the Database Application Tables (Oracle) connector will attempt to match on the column names. If the column names match then the relationship is matched. For example:

```
CREATE TABLE MYDBAT_PERSON
  (USERID VARCHAR2 NOT NULL ENABLE,
   USERNAME VARCHAR2 NOT NULL ENABLE,
```

```
   FIRSTNAME VARCHAR2,
   LASTNAME VARCHAR2,
   EMAIL VARCHAR2 NOT NULL ENABLE,
   COUNTRYCODE VARCHAR2);

CREATE TABLE MYDBAT_COUNTRY
  (COUNTRYCODE VARCHAR2 NOT NULL ENABLE,
   COUNTRYNAME VARCHAR2 NOT NULL ENABLE,
   CONSTRAINT MYDBAT_COUNTRY_PK PRIMARY KEY (COUNTRYCODE)
```

However, if there is no foreign key relationship defined between the tables, and the column names do not match, then an error will be thrown. This would happen, for example, if you define the columns as below:

```
CREATE TABLE MYDBAT_PERSON
  (USERID VARCHAR2 NOT NULL ENABLE,
   USERNAME VARCHAR2 NOT NULL ENABLE,
   FIRSTNAME VARCHAR2,
   LASTNAME VARCHAR2,
   EMAIL VARCHAR2 NOT NULL ENABLE,
   HOMECOUNTRY VARCHAR2);

CREATE TABLE MYDBAT_COUNTRY
  (COUNTRYCODE VARCHAR2 NOT NULL ENABLE,
   COUNTRYNAME VARCHAR2 NOT NULL ENABLE,
   CONSTRAINT MYDBAT_COUNTRY_PK PRIMARY KEY (COUNTRYCODE)
```

The same principles as explained above for the relationship between a user and a lookup would apply when defining tables for the relationship between a user and entitlements.

On creation of a Database Application Tables (Oracle) orchestrated system, the agent creates a `schema.json` in the agent directory (`<agent-volume>/data/schema`) which maps identity data between the integrated database tables and Oracle Access Governance. This file can be accepted as-is, or modified by you to configure what entities and attributes are mapped to Oracle Access Governance for consideration in campaigns and reviews.

Schema Discovery Flow

The schema discovery flow for Database Application Tables (Oracle) is as follows:

1. **Day0**: When you create the Database Application Tables (Oracle) orchestrated system, the initial `schema.json` file is created, based on the integration settings you enter. The schema file is created with details of tables and columns containing user data for Account, Entitlement, and Lookup entities. Relationship information is loaded from the configuration you supply, together with any constraints set in the database (Primary and Foreign keys). The intermediate schema JSON file will contain mapping for the following attributes by default, if they are defined. To map additional attributes you should edit the `schema.json` file:
   - **UID**: `__UID__`
   - **Name**: `__NAME__`

- **Status**: `__ENABLE__`
- **Password**: `__PASSWORD__`

2. **DayN**: DayN activities take place at any point after you have generated the intermediate schema JSON file. The default file created by the agent can be used for testing, but does not support full data load since it only brings UID and name attributes. For full data load you should edit the file to determine which entities and attributes get passed to Oracle Access Governance. For full details on the structure and options available when editing the `schema.json`, refer to Schema JSON File Reference.

> **Note:**
>
> Post-Day0 you cannot rerun schema discovery against the source database and generate a new schema file. The schema JSON file located with your agent is the source of truth for DayN activities, and any updates must be made in the file.

DayN activities may include the following:

a. **Run dataload with the limited schema**. You can run a test dataload with the limited schema generated on Day0. This will only include, where available, the attributes uid, name, status, and password.

b. **Modify schema file**: You can modify the `schema.json` file to update Account, Entitlement, and Lookup details if required. Some reasons why you might do this are:

   - You want to onboard attributes additional to the default uid, name, status, and password ones. To do this, set the `name` property for any attributes you want to be included.
   - If there is a change in the source database then this should be reflected in the schema file. For example, if you add a new column into the ACCOUNT table, that would need to be added to the schema file as an attribute.

c. You should perform a schema discovery operation which will fetch the latest custom attribute information. For details on how to perform this task, see Fetch Latest Custom Attributes.

d. **Run data load and provisioning operations**: Using the discovered schema, load user data into Oracle Access Governance from the integrated database, and allow Oracle Access Governance to provision accounts and permissions in the integrated database.

Intermediate Schema JSON

The intermediate `schema.json` file is generated using all information available from the configuration you supply during orchestrated system creation. This serves as a template to which you can make any changes that affect the user data you can onboard. The structure of the intermediate `schema.json` file is as follows:

```
{
 "schemaTemplates":[
    {
        "type": "", // Type of entity "ACCOUNT", "ENTITLEMENT",
"TARGETACCOUNT" or "LOOKUP"
        "name": "", // Name of entity
        "displayName": "", // display name of entity
```

```
        "data": {
            // Key-value pairs representing static lookup data if any, or
else it will be missing from here.
        }
        "attributes": [
            {
                "name": "", // AG side Name of attribute
                "targetName": "", // Target side name of attribute
                "displayName": "" // Optional display name for the attribute
which will be given priority if provided, else attribute name will be used
for display name.
                "dataType": "", // Either of TEXT, DATE, NUMBER,
DECIMAL_NUMBER, FLAG
                "nature": [ // One or more of "REQUIRED", "MULTIVALUED",
"SENSITIVE". It can be missing from here if nothing applies.
                ],
                "usage": [ // One or more of "READ", "PROVISION". It can be
missing from here if nothing applies.
                ],
                "relationship": { // Entity relationship details
                    "relatedTo": "", // Entity name in relationship with
                    "relatedBy": "", // Attribute to define the relation
                    "relationshipProperties": [ // Additional relationship
properties
                        {
                            "name": "", // Name of additional attribute
                            "dataType": "", // Either of TEXT, DATE, NUMBER,
DECIMAL_NUMBER, FLAG
                            "nature": [ // Only READ_ONLY is possible, or
else it will be missing from here
                            ],
                            "uiProperties": { // ARMD if applicable, or it
will be missing from here
                                "inputType": "" // Either of Auto, User, Admin
                                "widget": "", // Widget to use on UI i.e.
Either of Text, Password, Number, Date, SelectOne, RepeatableFieldSet,
CheckboxSet
                                "title": "", // Title to use on UI
                                "labelHint": "", // Labelhint to use on UI
                                "minLength": {SOME_POSITIVE_NUMBER},
                                "maxLength": {SOME_POSITIVE_NUMBER},
                                "defaultValues": [ // Default values if
applicable, or it will be missing from here
                                ]
                            }
                        }
                    ]
                },
                "outboundTransformation": { // Outbound transformation script
if applicable, or it will be missing from here
                    "script": "" // Script to execute for transformation
                },
                "uiProperties": { // ARMD if applicable, or it will be
missing from here
                    "inputType": "" // Either of Auto, User, Admin
                    "widget": "", // Widget to use on UI i.e. Either of Text,
```

```
Password, Number, Date, SelectOne, RepeatableFieldSet, CheckboxSet
                    "title": "", // Title to use on UI
                    "labelHint": "", // Labelhint to use on UI
                    "minLength": {SOME_POSITIVE_NUMBER},
                    "maxLength": {SOME_POSITIVE_NUMBER},
                    "defaultValues": [ // Default values if applicable, or it
will be missing from here
                    ]
                }
            }
        ]
    }
  ]
}
```

> **Note:**
>
> The `uiProperties` attribute is only in scope for **TARGETACCOUNT** entity types. It should not be used for ACCOUNT, ENTITLEMENT, or LOOKUP entity types.

The schema can contain entities from the following types:

- **ACCOUNT**: The ACCOUNT entity type maps to the Oracle Access Governance **Identity** entity. Attributes from the ACCOUNT entity type are used to populate an Identity within Oracle Access Governance.
- **TARGETACCOUNT**: The TARGETACCOUNT entity type maps Oracle Access Governance **Account** entity. When you provision an account in your database application, attributes from the ACCOUNT Oracle Access Governance entity type will be used to populate the TARGETACOUNT database entity.
- **ENTITLEMENT**: ENTITLEMENT maps to Oracle Access Governance **Permission**.
- **LOOKUP**: LOOKUP maps to **Lookup** information in Oracle Access Governance.

When the intermediate schema JSON is generated, it populates both ACCOUNT and TARGETACCOUNT for each user in the database tables. This happens regardless of whatever mode you configure the orchestrated system, authoritative source and/or managed system. However, when the dataload is run, then the following rules apply regarding which of these entities are actually created in Oracle Access Governance.

1. Mode selected is **Authoritative Source** only:
   - ACCOUNT and TARGETACCOUNT are generated in the schema JSON.
   - Only an Identity is created in Oracle Access Governance, based on the values of ACCOUNT.
   - TARGETACCOUNT can be ignored.
2. Mode selected is **Managed System** only:
   - ACCOUNT and TARGETACCOUNT are generated in the schema JSON.
   - Only an Account is created in Oracle Access Governance, based on the values of TARGETACCOUNT.
   - ACCOUNT can be ignored.
3. Mode selected is **Authoritative Source** and **Managed System**.

- ACCOUNT and TARGETACCOUNT are generated in the schema JSON.
- Identity is created in Oracle Access Governance, based on the values of ACCOUNT.
- Account is created in Oracle Access Governance, based on the values of TARGETACCOUNT.

> **Note:**
>
> Any entitlements that are granted directly in the database tables, that are then loaded into Oracle Access Governance cannot be managed by Oracle Access Governance when the orchestrated system is configured in managed systems mode. Only entitlements that originate in a provisioning request from Oracle Access Governance can be managed by Oracle Access Governance.

Core Identity Attributes

When you run Database Application Tables (Oracle) in authoritative source mode, you can onboard user data from your integrated database into Oracle Access Governance, which is used to create an Oracle Access Governance Identity.

The list of core attributes that can be assigned to an Oracle Access Governance Identity are shown below. The element, `"nature": ["REQUIRED"]` indicates that an attribute must be NOT NULL in the source database. This does not mean that the attribute must be included in the schema discovery, you may or may not include it into the schema you want to integrate.

```
[
  {
    "name": "uid",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "name",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "email",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
```

```
    ]
  },
  {
    "name": "firstName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "middleName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "lastName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "displayName",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "employeeType",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "title",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "empNo",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "status",
```

```
      "dataType": "FLAG",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "jobCode",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "state",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "risk",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "location",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "department",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "managerUid",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "managerLogin",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "organizationUid",
```

```
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "organizationName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "country",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "postalCode",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "territory",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  }
]
```

User Status

When you run Database Application Tables (Oracle) in authoritative source mode, you can onboard user data from your integrated database into Oracle Access Governance, which is used to create an Oracle Access Governance Identity.

When you configure your orchestrated system you are prompted for the database column holding the status of a user record. This column determines whether the user is enabled or disabled in Oracle Access Governance. This information is always captured during full data load, irrespective of the configured mode (authoritative source or managed system).

You also configure the following boolean values for enabled/disabled status:

- **User account enabled status value**: This value will default to ACTIVE or can be defined as any text value from the database which maps to a status of enabled in Oracle Access Governance. Some examples include *Y*, *Yes*, *true* and so on.
- **User account disabled status value**: This value will default to INACTIVE or can be defined as any text value from the database which maps to a status of disabled in Oracle Access Governance. Some examples include *N*, *No*, *false* and so on.

Mapping from the database value to Oracle Access Governance enabled/disabled status uses the following logic:

- If the user has a status which equals the **User account disabled status value** value then the user will be marked as disabled in Oracle Access Governance.
- If the user has any other value then it is assumed that the user is enabled and the user is marked as such in Oracle Access Governance.

Updating the Intermediate Schema JSON File

You can update the intermediate schema JSON file to include attributes in the Oracle Access Governance schema.

To include or exclude Oracle Access Governance schema attributes, consider the following example entity, which shows an example of an ACCOUNT entity:

```
"type" : "ACCOUNT",
"name" : "ACCOUNT",
"displayName" : "Account",
"attributes" : [ {
  "name" : "joiningDate",
  "targetName" : "JOININGDATE",
  "displayName" : "",
  "dataType" : "DATE",
  "nature" : [ ],
  "usage" : [ "READ" ]
}, {
  "name" : "",
  "targetName" : "LASTUPDATED",
  "displayName" : "",
  "dataType" : "DATE",
  "nature" : [ ],
  "usage" : [ "READ" ]
}, {
  "name" : "firstName",
  "targetName" : "FIRSTNAME",
  "displayName" : "",
  "dataType" : "TEXT",
  "nature" : [ ],
  "usage" : [ "READ" ]
}, {
  "name" : "email",
  "targetName" : "EMAIL",
  "displayName" : "",
  "dataType" : "TEXT",
  "nature" : [ "REQUIRED" ],
  "usage" : [ "READ" ]
}, {
  "name" : "name",
  "targetName" : "__NAME__",
  "displayName" : "",
  "dataType" : "TEXT",
  "nature" : [ "REQUIRED" ],
  "usage" : [ "READ" ]
}, {
  "name" : "uid",
```

```
      "targetName" : "__UID__",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ "REQUIRED" ],
      "usage" : [ "READ" ]
    }, {
      "name" : "status",
      "targetName" : "__ENABLE__",
      "displayName" : "",
      "dataType" : "FLAG",
      "nature" : [ ],
      "usage" : [ "READ" ]
    }, {
      "name" : "salary",
      "targetName" : "SALARY",
      "displayName" : "",
      "dataType" : "DECIMAL_NUMBER",
      "nature" : [ ],
      "usage" : [ "READ" ]
    }, {
      "name" : "password",
      "targetName" : "__PASSWORD__",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ "SENSITIVE" ],
      "usage" : [ ]
    }, {
      "name" : "country",
      "targetName" : "COUNTRYCODE",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ ],
      "usage" : [ "READ" ],
      "relationship" : {
        "relatedTo" : "A_COUNTRY",
        "relatedBy" : "COUNTRYCODE"
      }
    }, {
      "name" : "",
      "targetName" : "DESCRIPTION",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ ],
      "usage" : [ "READ" ]
    }, {
      "name" : "lastName",
      "targetName" : "LASTNAME",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ ],
      "usage" : [ "READ" ]
    } ]
  }, {
```

Including Attributes in your Schema

Note the `name` and `targetName` parameters for the attributes listed in the example. You will see that in all cases the `targetName` parameter is populated. This is because this parameter maps to the column name in your Oracle database which contains the user information you are integrating. So, for example, we have `FIRSTNAME`, `LASTNAME`, `DESCRIPTION`, and so on. The `name` parameter is, however, not always populated. This is because the `name` parameter correlates to the attribute in the schema on the Oracle Access Governance side. If `name` is populated, then that parameter is included in the Oracle Access Governance schema, for example:

```
"name" : "firstName",
"targetName" : "FIRSTNAME",
```

If `name` is not populated, then that parameter is not included in the Oracle Access Governance schema. In the example above the parameter `DESCRIPTION` is not included:

```
"name" : "",
"targetName" : "DESCRIPTION",
```

Including Lookup Tables

To include lookup tables to your schema, you use the same method as above, by setting the `name` parameter in your schema JSON file. Setting the `name` parameter will include any lookup data and will map it to the relevant Oracle Access Governance attribute. Additionally, if you include a lookup table, the values will be displayed in a list-of-values when creating a new access bundle.

```
    "type" : "LOOKUP",
    "name" : "COUNTRY",
    "displayName" : "Country",
    "attributes" : [ {
      "name" : "uid",
      "targetName" : "__UID__",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ "REQUIRED" ],
      "usage" : [ "READ" ]
    }, {
      "name" : "name",
      "targetName" : "__NAME__",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ "REQUIRED" ],
      "usage" : [ "READ" ]
    } ]
  }
```

Additionally, you should create a field in your main ACCOUNT table which holds the value from your lookup. In this example you might have an attribute something like `account.homeCountry`

which holds this value. This attribute should have a foreign key relationship with your ACCOUNT table. An example might be:

```
    "type" : "ACCOUNT",
    "name" : "ACCOUNT",
    "displayName" : "Account",
    "attributes" : ...
{
      "name" : "country",
      "targetName" : "COUNTRYCODE",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ ],
      "usage" : [ "READ" ],
      "relationship" : {
        "relatedTo" : "COUNTRY",
        "relatedBy" : "COUNTRYCODE"
      }
```

Mapping to Core and Custom Attributes

You can determine if a parameter is mapped to a core attribute or a custom attribute. If you use the core attribute name in the `name` then the corresponding `targetName` value is mapped to the core attribute. If `name` is not the name of a core attribute, then the `targetName` value is mapped to a custom attribute. For example the following would map the database column `FIRSTNAME` to the core attribute `firstName`.

```
{
      "name" : "firstName",
      "targetName" : "FIRSTNAME",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ ],
      "usage" : [ "READ" ]
    }
```

while the following would map `FIRSTNAME` to a custom attribute called `foreName`

```
{
      "name" : "foreName",
      "targetName" : "FIRSTNAME",
      "displayName" : "",
      "dataType" : "TEXT",
      "nature" : [ ],
      "usage" : [ "READ" ]
    }
```

.

Custom Scripting for Overview

When you provision accounts from Oracle Access Governance using the Database Application Tables (Oracle) integration, operations such as create, update, and delete are implemented using the default supplied code. On occasions where you wish to modify the default supplied operations, you can optionally provide your own custom scripts which implement your own specific provisioning operation requirements. This step is completely optional, you do not have to create custom scripts if the default operations provide you with what you need. You can add custom scripts to any operations supported. If you choose custom scripts, you only need to add them where you require the default operation to be modified, you can have a combination of custom and default scripts for the operations supported, though you can only have one or the other option for each specific operation. For example, the *create* operation might be implemented with a custom script that adds some functionality specific to your organization, while the *delete* operation is unchanged and uses default functionality.

Once you have implemented and configured your Database Application Tables (Oracle) to use a custom script then that script will be used when you next perform a provisioning or data load operation.

> **Note:**
>
> Any custom script must be implemented using Groovy format. Other scripting formats are not supported.

When you create a Database Application Tables (Oracle) orchestrated system you can identify scripts to be run for a number of provisioning operations on the database application containing account data. These operations are:

- Create
- Update
- Delete
- Dataload
- Add relationship data
- Remove relationship data

These scripts should be located on the agent host, in the install directory of the agent, for example, `/app/<custom script>` . You configure the agent with the location of the scripts in the integration settings for your orchestrated system. You should ensure that the operating system user running the agent has read/write permissions for any custom scripts.

When you perform a provisioning task, your script will be run as a replacement to the standard processing associated with the task. The script must handle the default provisioning task such as create or update, and can also have custom tasks above and beyond the default provisioning process, such as:

- Perform custom table updates
- Custom auditing
- Send custom notifications

This means that you have two options for provisioning processing using the Database Application Tables (Oracle) integration:

1. Use the default logic provided with the Database Application Tables (Oracle) connector

2. Use the custom logic implemented in scripts

Custom scripts are only used when configured in your orchestrated system. So, if you have specified a create script when creating your orchestrated system, but no script for update, then the custom script will be used for the create provisioning task, while the update task will be implemented using the default connector processing.

You should also note that all custom script types are supported for an orchestrated system configured for managed system mode. The only script type supported for authoritative source mode is the Dataload type, which is supported for both modes.

> **Note:**

Sample Database Schema

The samples provided in the following sections are based on the database tables described in this section.

MYDBAT_PERSON

```
CREATE TABLE MYDBAT_PERSON
  (USERID VARCHAR2(50BYTE) NOT NULL ENABLE,
   USERNAME VARCHAR2(50BYTE) NOT NULL ENABLE,
   FIRSTNAME VARCHAR2(50BYTE),
   LASTNAME VARCHAR2(50BYTE),
   EMAIL VARCHAR2(50BYTE) NOT NULL ENABLE,
   COUNTRYCODE VARCHAR2(20BYTE),
   DESCRIPTION VARCHAR2(50BYTE),
   SALARY NUMBER,
   JOININGDATE DATE,
   STATUS VARCHAR2(50BYTE),
   LASTUPDATED TIMESTAMP (6),
   PASSWORD VARCHAR2(50BYTE),
   CONSTRAINT MYDBATPERSON_PK PRIMARY KEY (USERID));
```

MYDBAT_GROUPS

```
CREATE TABLE MYDBAT_GROUPS
  (GROUPID VARCHAR2(20BYTE) NOT NULL ENABLE,
   GROUPNAME VARCHAR2(20BYTE) NOT NULL ENABLE,
   CONSTRAINT MYDBATGROUPS_PK PRIMARY KEY (GROUPID));
```

MYDBAT_ROLES

```
CREATE TABLE MYDBAT_ROLES
  (ROLEID VARCHAR2(50BYTE) NOT NULL ENABLE,
   ROLENAME VARCHAR2(50BYTE) NOT NULL ENABLE,
   CONSTRAINT MYDBATROLES_PK PRIMARY KEY (ROLEID));
```

MYDBAT_PERSON_GROUP

```
CREATE TABLE MYDBAT_PERSON_GROUP
  (USERID VARCHAR2(20BYTE) NOT NULL ENABLE,
   GROUPID VARCHAR2(20BYTE) NOT NULL ENABLE,
   CONSTRAINT "MYDBATPERSONGROUP_PK"PRIMARY KEY (USERID, GROUPID));
```

MYDBAT_PERSON_ROLE

```
CREATE TABLE MYDBAT_PERSON_ROLE
  (USERID VARCHAR2(20BYTE) NOT NULL ENABLE,
   ROLEID VARCHAR2(20BYTE) NOT NULL ENABLE,
   FROMDATE DATE,
   TODATE DATE,
   CONSTRAINT "MYDBATPERSONROLE_PK"PRIMARY KEY (USERID, ROLEID));
```

MYDBAT_COUNTRY

```
CREATE TABLE MYDBAT_COUNTRY
  (COUNTRYCODE VARCHAR2(20BYTE) NOT NULL ENABLE,
   COUNTRYNAME VARCHAR2(200BYTE) NOT NULL ENABLE,
   CONSTRAINT MYDBAT_COUNTRY_PK PRIMARY KEY (COUNTRYCODE));
```

> **Note:**
>
> Child tables such as `mydbat_roles`, `mydbat_groups`, and `mydbat_country` should have a primary key constraint defined. If no primary key is defined for child tables then your validate operation will fail and you will see an error **Key for table <tablename> are not defined**.

Trigger and Sequence For Auto-Increment

The following code will create a sequence and trigger which will auto-assign a USERID when creating a user in the MYDBAT_PERSON table.

```
CREATE SEQUENCE MYDBAT_PERSON_SEQ MINVALUE 1 MAXVALUE
9999999999999999999999999999 INCREMENT
      BY 1 START WITH 1 CACHE 20 NOORDER NOCYCLE;


create or replace TRIGGER DBAT_PERSON_TRG
BEFORE INSERT ON MYDBAT_PERSON
FOR EACH ROW
BEGIN
  <<COLUMN_SEQUENCES>>
  BEGIN IF INSERTING AND :NEW.USERID IS NULL THEN
    SELECT MYDBAT_PERSON_SEQ.NEXTVAL INTO :NEW.USERID FROM SYS.DUAL;
  END IF;
  END COLUMN_SEQUENCES;END;
```

Groovy Script Arguments

The following arguments can be used in your Groovy scripts:

Table 5-19 Script Arguments

Argument	Description
connector	The Database Application Tables connector object.

timing	When the Groovy script is called. The timing attribute also explains the type of operation being performed. For example, if it is a search operation, then the object class being searched is also returned.
	The following is the format of the timing argument for lookup field synchronization:
	`executeQuery:OBJECT_CLASS`
	In this format `OBJECT_CLASS` is replaced with the type of object being reconciled.
	For example, for a lookup field synchronization scheduled job that contains the object type *Role*, the value of the timing argument will be as follows:
	`executeQuery:Role`

attributes	All attributes.
trace	Logger as a script trace bridge to the application
where	String where condition for execute query, or null.
handler	resultSetHandler or SyncResultsHandler for the connector objects produced by the execute query, sync operation or null return.
quoting	The type of table name quoting to be used in SQL. The default value is an empty string. The value of this argument is obtained from the integration settings.
nativeTimestamps	Specifies whether the script retrieves the timestamp data of the columns as java.sql.Timestamp type from the database table. This information is obtained from the integration settings.
allNative	Specifies whether the script must retrieve the data type of the columns in a native format from the database table. The value of this argument is obtained from the integration settings. The value of this argument specifies whether the script must throw exceptions when a zero (0x00) error code is encountered.
enableEmptyString	Specifies whether support for writing an empty string instead of a NULL value must be enabled. The value of this argument is obtained from the integration settings.
filterString	String filter condition for execute query, or null.

Table 5-19 (Cont.) Script Arguments

Argument	Description
filterParams	List of filter parameters. Each parameter is present in the COLUMN_NAME:VALUE format. For example, FIRSTNAME:test.
syncattribute	Name of the database column configured for incremental reconciliation. This argument is available in the sync script, which is called during an incremental reconciliation run.
synctoken	Value of the sync attribute. This argument is available in the sync script.

Sample Dataload Script

The data load script reads the data from all the tables for all the defined entities. In this scenario, the term data load refers to the full data load and the lookup data load.

This sample script reads user data from the MYDBAT_PERSON table, and the users' relationship data from the MYDABAT_PERSON_ROLE and MYDBAT_PERSON_GROUP tables. Entitlements data is read from the MYDBAT_GROUPS table, and lookup data is read from the MYDBAT_COUNTRY table. It also has support for a basic filter search on MYDBAT_PERSON table. All these data reads are done using stored procedures.

Dataload Script

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.math.*;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Map;
import org.identityconnectors.framework.common.objects.*;
import java.lang.reflect.*;
import java.lang.String;
import org.identityconnectors.common.security.GuardedString;
import java.text.*;

ResultSet rs = null;
        CallableStatement st = null;
        String ocName ;
        try {
            if( timing != "")
            {
                trace.info("[Execute Query] timing attribute value : "+
timing);
                ocName = timing.split(":")[1]
            }

            trace.info("[Execute Query] for objectClass : "+ ocName);

            if(ocName.equals("ACCOUNT") || ocName.equals("TARGETACCOUNT")){
```

```
            if( filterString != "")
            {
                trace.info("[Execute Query] Performing Recon with Filter.
Filter is::"+ filterString+" And Filer Params are::"+filterParams);
                //[Execute Query] Performing Recon with Filter. Filter
is::MYDBAT_PERSON.USERID = ? And Filer Params are::[MYDBAT_PERSON.USERID:21]
                String[] filter = filterParams.get(0).split(":");
                st = conn.prepareCall("{call
EXECUTE_QUERY_WITH_FILTER(?,?,?)}");
                st.setString(2, filter[0]);
                st.setString(3, filter[1]);
            }
            else
            {
                trace.info("[Execute Query] Performing Full Recon.");
                st = conn.prepareCall("{call EXECUTE_QUERY_PERSON(?)}");
            }
            st.registerOutParameter(1, oracle.jdbc.OracleTypes.CURSOR);
            st.execute();
            rs = (ResultSet) st.getObject(1);
            SimpleDateFormat targetFormat = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss z");
            DateFormat df = new SimpleDateFormat("yyyy-MM-dd");

            while (rs.next()) {
                ConnectorObjectBuilder cob = new ConnectorObjectBuilder();
                cob.setObjectClass(ObjectClass.ACCOUNT);
                Attribute fname= AttributeBuilder.build(new
String("FIRSTNAME"),rs.getString(2));
                Attribute lname= AttributeBuilder.build(new
String("LASTNAME"),rs.getString(3));
                Attribute uid= AttributeBuilder.build(new
String("__UID__"),rs.getString(1));
                Attribute name= AttributeBuilder.build(new
String("__NAME__"),rs.getString(10));
                Attribute email= AttributeBuilder.build(new
String("EMAIL"),rs.getString(4));
                //Attribute salary= AttributeBuilder.build(new
String("SALARY"),rs.getBigDecimal(6));
                Attribute description= AttributeBuilder.build(new
String("DESCRIPTION"),rs.getString(5));
                Date dbDate = rs.getDate(7);
                String joinDateStr = null;
                Long joinDate = null;
                if( null != dbDate)
                {
                    java.util.Date date= df.parse(dbDate.toString());
                    joinDateStr = targetFormat.format(date);
                    joinDate = date.getTime()
                    trace.info("date : " +date +" ---- joinDate : "+
joinDate);
                }

                //Attribute joindate= AttributeBuilder.build(new
String("JOININGDATE"),joinDateStr);
                if(null != joinDate) {
```

```
                    trace.info("Setting joinDate : "+ joinDate);
                    Attribute joindate= AttributeBuilder.build(new
String("JOININGDATE"),joinDate);
                    cob.addAttribute(joindate);
                }
                Attribute status= AttributeBuilder.build(new
String("STATUS"),rs.getString(8));
                Attribute countryCode= AttributeBuilder.build(new
String("COUNTRYCODE"),rs.getString(9));
                cob.addAttribute(fname);
                cob.addAttribute(lname);
                cob.addAttribute(uid);
                cob.addAttribute(name);
                cob.addAttribute(email);
                //cob.addAttribute(salary);
                cob.addAttribute(description);
                cob.addAttribute(status);
                cob.addAttribute(countryCode);

                if(ocName.equals("TARGETACCOUNT")){
                CallableStatement roleStmt = conn.prepareCall("{call
GET_USERROLE(?,?)}");
                roleStmt.registerOutParameter(1,
oracle.jdbc.driver.OracleTypes.CURSOR);
                roleStmt.setString(2, rs.getString(1));
                roleStmt.execute();
                ResultSet roleResultSet = (ResultSet) roleStmt.getObject(1);
                java.util.List<EmbeddedObject> eoList = new
ArrayList<EmbeddedObject>();
                while (roleResultSet.next()) {
                    Attribute roleId= AttributeBuilder.build(new
String("ROLEID"),roleResultSet.getString(2));
                    dbDate = roleResultSet.getDate(3);
                    String fromDateStr = null;
                    Long fromDate = null;
                    if( null != dbDate)
                    {
                        java.util.Date date= df.parse(dbDate.toString());
                        fromDateStr = targetFormat.format(date);
                        fromDate = date.getTime()
                    }

                    dbDate = roleResultSet.getDate(4);
                    String toDateStr = null;
                    Long toDate = null;
                    if( null != dbDate)
                    {
                        java.util.Date date= df.parse(dbDate.toString());
                        toDateStr = targetFormat.format(date);
                        toDate = date.getTime()
                    }

                    EmbeddedObjectBuilder roleEA = new
EmbeddedObjectBuilder();
                    roleEA.addAttribute(roleId);
                    if(null != fromDate) {
```

```
                        trace.info("Setting roles fromDate : "+ fromDate);
                        Attribute fromdate= AttributeBuilder.build(new
String("FROMDATE"),fromDate);
                        roleEA.addAttribute(fromdate);
                    }

                    if(null != toDate) {
                        trace.info("Setting roles toDate : "+ toDate);
                        Attribute todate= AttributeBuilder.build(new
String("TODATE"),toDate);
                        roleEA.addAttribute(todate);
                    }

                    roleEA.setObjectClass(new ObjectClass("MYDBAT_ROLES"));
                    eoList.add(roleEA.build());
                }
                roleResultSet.close();
                roleStmt.close();
                EmbeddedObject[] roleEm = eoList.toArray(new
EmbeddedObject[eoList.size()]);
                cob.addAttribute(AttributeBuilder.build("MYDBAT_PERSON_ROLE",
(Object[]) roleEm));

                CallableStatement groupStmt = conn.prepareCall("{call
GET_USERGROUP(?,?)}");
                groupStmt.registerOutParameter(1,
oracle.jdbc.driver.OracleTypes.CURSOR);
                groupStmt.setString(2, rs.getString(1));
                groupStmt.execute();
                ResultSet groupResultSet = (ResultSet) groupStmt.getObject(1);
                java.util.List<EmbeddedObject> geoList = new
ArrayList<EmbeddedObject>();
                while (groupResultSet.next()) {
                    Attribute groupId= AttributeBuilder.build(new
String("GROUPID"),groupResultSet.getString(2));
                    EmbeddedObjectBuilder groupEA = new
EmbeddedObjectBuilder();
                    groupEA.addAttribute(groupId);
                    groupEA.setObjectClass(new ObjectClass("MYDBAT_GROUPS"));
                    geoList.add(groupEA.build());
                }
                groupResultSet.close();
                groupStmt.close();
                EmbeddedObject[] groupEm = geoList.toArray(new
EmbeddedObject[geoList.size()]);

cob.addAttribute(AttributeBuilder.build("MYDBAT_PERSON_GROUP", (Object[])
groupEm));
                }

                if(!handler.handle(cob.build())) return;
            }
    }else if(ocName.equals("MYDBAT_COUNTRY")){
        trace.info("[Execute Query] for Lookup : "+ ocName);
        CallableStatement groupStmt = conn.prepareCall("{call
GET_COUNTRIES(?)}");
```

```
        groupStmt.registerOutParameter(1,
oracle.jdbc.driver.OracleTypes.CURSOR);
        groupStmt.execute();
        ResultSet groupResultSet = (ResultSet) groupStmt.getObject(1);
        while (groupResultSet.next()) {
            ConnectorObjectBuilder cob = new ConnectorObjectBuilder();
            cob.setObjectClass(new ObjectClass("MYDBAT_COUNTRY"));
            Attribute groupId= AttributeBuilder.build(new
String("__UID__"),groupResultSet.getString(1));
            Attribute groupName= AttributeBuilder.build(new
String("__NAME__"),groupResultSet.getString(2));
            cob.addAttribute(groupId);
            cob.addAttribute(groupName);
            if(!handler.handle(cob.build())) return;
        }

        groupResultSet.close();
        groupStmt.close();
    }else if(ocName.equals("MYDBAT_GROUPS")){
        trace.info("[Execute Query] for Entitlement : "+ ocName);
        CallableStatement groupStmt = conn.prepareCall("{call
GET_GROUPS(?)}");
        groupStmt.registerOutParameter(1,
oracle.jdbc.driver.OracleTypes.CURSOR);
        groupStmt.execute();
        ResultSet groupResultSet = (ResultSet) groupStmt.getObject(1);
        while (groupResultSet.next()) {
            ConnectorObjectBuilder cob = new ConnectorObjectBuilder();
            cob.setObjectClass(new ObjectClass("MYDBAT_GROUPS"));
            Attribute groupId= AttributeBuilder.build(new
String("__UID__"),groupResultSet.getString(1));
            Attribute groupName= AttributeBuilder.build(new
String("__NAME__"),groupResultSet.getString(2));
            cob.addAttribute(groupId);
            cob.addAttribute(groupName);
            if(!handler.handle(cob.build())) return;
        }

        groupResultSet.close();
        groupStmt.close();
    }
    } finally {
            if( null != rs)
                rs.close();
            if( null != st)
                st.close();
        }
```

Stored Procedure: Load Users

```
create or replace PROCEDURE EXECUTE_QUERY_PERSON
(user_cursor OUT SYS_REFCURSOR) AS
BEGIN
  OPEN user_cursor FOR
  SELECT USERID,
```

```
          FIRSTNAME,
          LASTNAME,
          EMAIL,
          DESCRIPTION,
          SALARY,
          JOININGDATE,
          STATUS,
          COUNTRYCODE,
          USERNAME
  FROM MYDBAT_PERSON;
END EXECUTE_QUERY_PERSON;
```

Stored Procedure: Filtered User Search

```
create or replace PROCEDURE EXECUTE_QUERY_WITH_FILTER
(user_cursor OUT SYS_REFCURSOR,
 filter IN VARCHAR2,
 filterValue IN VARCHAR2) AS
BEGIN
  OPEN user_cursor FOR
    SELECT USERID,
           FIRSTNAME,
           LASTNAME,
           EMAIL,
           DESCRIPTION,
           SALARY,
           JOININGDATE,
           STATUS,
           COUNTRYCODE,
           USERNAME
    FROM MYDBAT_PERSON
    WHERE filter=filterValue;
END EXECUTE_QUERY_WITH_FILTER;
```

This is a very basic example of filter search with only one filter condition, for example, *MYDBAT_PERSON.USERID:21*. This is used specifically for writeBack processing after the create operation

Stored Procedure: Get Roles

```
create or replace PROCEDURE GET_ROLES
(user_cursor OUT SYS_REFCURSOR) AS
BEGIN
  OPEN user_cursor FOR
    SELECT ROLEID,
           ROLENAME
    FROM MYDBAT_ROLES;
END GET_ROLES;
```

Stored Procedure: Get User Roles

```
create or replace PROCEDURE GET_USERROLE
(user_cursor OUT SYS_REFCURSOR,
 userin IN VARCHAR2) AS
```

```
BEGIN
  OPEN user_cursor FOR
    SELECT USERID,
           ROLEID,
           FROMDATE,
           TODATE
    FROM MYDBAT_PERSON_ROLE
    WHERE USERID=userin;
END GET_USERROLE;
```

Stored Procedure: Get Groups

```
create or replace PROCEDURE GET_GROUPS
(user_cursor OUT SYS_REFCURSOR) AS
BEGIN
  OPEN user_cursor FOR
    SELECT GROUPID,
           GROUPNAME
    FROM MYDBAT_GROUPS;
END GET_GROUPS;
```

Stored Procedure: Get User Groups

```
create or replace PROCEDURE GET_USERGROUP
(user_cursor OUT SYS_REFCURSOR,
 userin IN VARCHAR2) AS
BEGIN
  OPEN user_cursor FOR
    SELECT USERID,
           GROUPID
    FROM MYDBAT_PERSON_GROUP
    WHERE USERID=userin;
END GET_USERGROUP;
```

Stored Procedure: Get Lookups (Country)

```
create or replace PROCEDURE GET_COUNTRIES
(user_cursor OUT SYS_REFCURSOR) AS
BEGIN
  OPEN user_cursor FOR
    SELECT COUNTRYCODE,
           COUNTRYNAME
    FROM MYDBAT_COUNTRY;
END GET_COUNTRIES;
```

Sample Create Script

This script is invoked during provisioning of a new account from Oracle Access Governance. Here we are inserting data into the MYDBAT_PERSON table.

Create Script

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.text.*;
import java.util.Date.*;
import org.identityconnectors.common.security.GuardedString;
import org.identityconnectors.framework.common.objects.*;

import java.text.*;

// START HERE
trace.info("[Create-Groovy] Attributes::"+attributes);
//
USERID,PASSWORD,USERNAME,STATUS,EMAIL,FIRSTNAME,LASTNAME,ORGANIZATION,CITY,EMP
LOYEE_NUMBER,joiningdate,ENDDATE,LONGVALUE,FLOATVALUE,CHARVALUE
//Get all the attributes from script argument

// This shows how to read arrtibures


String uid = attributes.get("__NAME__")!=null?
attributes.get("__NAME__").getValue().get(0):null;
GuardedString pass = attributes.get("__PASSWORD__")!=null?
attributes.get("__PASSWORD__").getValue().get(0):null;
String uname = attributes.get("__NAME__")!=null?
attributes.get("__NAME__").getValue().get(0):null;
enableValue = attributes.get("__ENABLE__")!=null?
attributes.get("__ENABLE__").getValue().get(0):true;
String email=attributes.get("EMAIL")!=null?
attributes.get("EMAIL").getValue().get(0):null;
String first=attributes.get("FIRSTNAME")!=null?
attributes.get("FIRSTNAME").getValue().get(0):null;
String last=attributes.get("LASTNAME")!=null?
attributes.get("LASTNAME").getValue().get(0):null;
String org=attributes.get("ORGANIZATION")!=null?
attributes.get("ORGANIZATION").getValue().get(0):null;
String countryCode=attributes.get("COUNTRYCODE")!=null?
attributes.get("COUNTRYCODE").getValue().get(0):null;
joiningdate = attributes.get("JOININGDATE")!=null?
attributes.get("JOININGDATE").getValue().get(0):null;


PreparedStatement createStmt = null;
String ret =null;
try {

        //Call Target API to create a user
        createStmt =   conn.prepareStatement("INSERT INTO
MYDBAT_PERSON(USERID,PASSWORD,USERNAME,STATUS,EMAIL,FIRSTNAME,LASTNAME,COUNTRY
```

```
CODE,JOININGDATE) VALUES(?,?,?,?,?,?,?,?,?)");
        createStmt.setString(1, uid);
        if(pass!=null)
        {
        pass.access(new GuardedString.Accessor(){
       public void access(char[] clearChars) {  createStmt.setString(2, new
String(clearChars));}
        });
        }
        else
        createStmt.setString(2,null);

        createStmt.setString(3, uname);

        if(enableValue)
                createStmt.setString(4,"Enabled");
        else
                createStmt.setString(4,"Disabled");
        createStmt.setString(5, email);
        createStmt.setString(6, first);
        createStmt.setString(7, last);
        createStmt.setString(8, countryCode);

        DateFormat formatter = new SimpleDateFormat("dd-MMM-yy");

        if (joiningdate!=null)
        {
            if (joiningdate == 0){
                createStmt.setString(9,null);
            }
            else
        {
        Date da=new Date(joiningdate);
        st=formatter.format(da);
        createStmt.setString(9,st);

        }
        }

       createStmt.executeUpdate();

} finally{
        //close the sql statements
        if (createStmt != null)
                createStmt.close();
}
trace.info("[Create] Created User::"+uid);
//Return Uid from the script
return new Uid(uid);
```

Sample Add Child Script

This script is invoked during provisioning of entitlements/permissions to users from Oracle Access Governance. Here we are inserting data into the MYDBAT_PERSON_GROUP and MYDBAT_PERSON_ROLE tables.

Add Child Script

```
import org.identityconnectors.framework.common.objects.*;
import java.text.*;

trace.info("[addMultiValuedAttributeScript-Groovy] Adding Child data::"+
attributes);
childst =null;
try {
        //Adding Group data

        childDataEOSet = null;

        //The child attributes are returned as a set of embedded objects.
Each  Embedded object
        // will provide a row of data in the child table.

        // Logic for handling simple multi valued attributes

        if(attributes.get("MYDBAT_PERSON_GROUP")!=null)
        {


childDataEOSet=attributes.get("MYDBAT_PERSON_GROUP").getValue();
                childst=conn.prepareStatement("INSERT INTO
MYDBAT_PERSON_GROUP VALUES (?,?)");
                String id = attributes.get("__UID__").getValue().get(0);

                if(childDataEOSet !=null){
                        //Iterate through child data and insert into table
                        trace.info("[addMultiValuedAttributeScript] Adding
Group data.");
                        for( iterator = childDataEOSet.iterator();
iterator.hasNext(); )
                        {
                                eo = iterator.next();
                                attrsSet=eo.getAttributes();

grpattr=AttributeUtil.find("GROUPID",attrsSet);
                                 if(grpattr!=null){
                                        // You are iterating simple multi
valued attributes here, Call target APIs here
                                        //conn object is available here
                                        groupid=grpattr.getValue().get(0);

                                        childst.setString(1, id);
                                        childst.setString(2, groupid);
                                        childst.executeUpdate();
                                        childst.clearParameters();
```

```
                                        }
                                };
                        }
                }
        } finally {
                if (childst != null)
                childst.close();

        };

        try {
                childDataEOSet = null;
                // Logic for handling Complex multi valued attributes
                if(attributes.get("MYDBAT_PERSON_ROLE")!=null)
                {


        childDataEOSet=attributes.get("MYDBAT_PERSON_ROLE").getValue();
                        childst=conn.prepareStatement("INSERT INTO MYDBAT_PERSON_ROLE
        VALUES (?,?,?,?)");

                        String id = attributes.get("__UID__").getValue().get(0);

                        if(childDataEOSet !=null)
                        {
                                trace.info("[addMultiValuedAttributeScript] Adding
        Role data.");
                                for( iterator = childDataEOSet.iterator();
        iterator.hasNext(); )
                                {
                                        eo = iterator.next();
                                        attrsSet = eo.getAttributes(); // Get all the
        attributes of child object

        roleattr=AttributeUtil.find("ROLEID",attrsSet);


                                        // You are iterating complex multi valued
        attributes here, Call target APIs here
                                    //conn object is available here
                                        if(roleattr!=null){
                                                // You are iterating simple multi
        valued attributes here, Call target APIs here
                                                //conn object is available here
                                                roleid=roleattr.getValue().get(0);

        fromDate=AttributeUtil.find("FROMDATE",attrsSet).getValue().get(0);

        toDate=AttributeUtil.find("TODATE",attrsSet).getValue().get(0);

                                                childst.setString(1, id);
                                                childst.setString(2, roleid);
                                                Date from_date=new Date(fromDate);
                                                SimpleDateFormat formatter = new
        SimpleDateFormat("dd-MMM-yy");
                                                String
```

```
from_date_st=formatter.format(from_date);
                                        childst.setString(3, from_date_st);

                                        Date to_date=new Date(toDate);
                                        String
to_date_st=formatter.format(to_date);
                                        childst.setString(4, to_date_st);


                                        childst.executeUpdate();
                                        childst.clearParameters();
                                }

                        };
                }
        }
} finally {
if (childst != null)
                childst.close();

};
```

Sample Remove Child Script

This script is invoked during deprovisioning of entitlements/permissions from users from Oracle Access Governance. Here we are removing data from MYDBAT_PERSON_GROUP and MYDBAT_PERSON_ROLE tables using stored procedures.

Remove Child Script

```
import org.identityconnectors.framework.common.objects.*;

trace.info("[removeMultiValuedAttributeScript] Removing Child data::"+
attributes);

try {
        childDataEOSet = null;
        delSt = null;
        //Get UID
        String id      = attributes.get("__UID__").getValue().get(0);
        if(attributes.get("MYDBAT_PERSON_GROUP")!=null)
        {

childDataEOSet=attributes.get("MYDBAT_PERSON_GROUP").getValue();
                //Delete child data using stored procedure
                delSt= conn.prepareCall("{call DELETE_USERGROUP(?,?)}");
            if(childDataEOSet !=null){
                        trace.info("[removeMultiValuedAttributeScript]
Removing Group data.");
                        //Iterate through child data and delete
                        for( iterator = childDataEOSet.iterator();
iterator.hasNext(); )
                        {
                                eo = iterator.next();
                                attrsSet = eo.getAttributes();
```

```
grpattr=AttributeUtil.find("GROUPID",attrsSet);
                                if(grpattr!=null){
                                        groupid=grpattr.getValue().get(0);
                                        delSt.setString(1, id);
                                        delSt.setString(2, groupid);
                                        delSt.executeUpdate();

trace.info("[removeMultiValuedAttributeScript] Deleted Group::"+ grpattr);
                                }
                        };
                }

        }
} finally {
        if (delSt != null)
        delSt.close();
};

try {
        childDataEOSet = null;
        delSt = null;
        String id      = attributes.get("__UID__").getValue().get(0);
        if(attributes.get("MYDBAT_PERSON_ROLE")!=null)
        {

childDataEOSet=attributes.get("MYDBAT_PERSON_ROLE").getValue();
                delSt= conn.prepareCall("{call DELETE_USERROLE(?,?)}");
            if(childDataEOSet !=null){
                        trace.info("[removeMultiValuedAttributeScript]
Removing Role data.");
                        for( iterator = childDataEOSet.iterator();
iterator.hasNext(); )
                        {

                                eo = iterator.next();
                                attrsSet = eo.getAttributes();

roleattr=AttributeUtil.find("ROLEID",attrsSet);
                                if(roleattr!=null){
                                        rolename=roleattr.getValue().get(0);
                                        delSt.setString(1, id);
                                        delSt.setString(2, rolename);
                                        delSt.executeUpdate();

trace.info("[removeMultiValuedAttributeScript] Deleted Role::"+ rolename);
                                }
                        };
                }
        }
} finally {
        if (delSt != null)
                delSt.close();
};
```

Stored Procedure: Remove Child

```
create or replace PROCEDURE DELETE_USERGROUP
(user_id MYDBAT_PERSON_group.USERID%TYPE,
 group_id MYDBAT_PERSON_group.GROUPID%TYPE ) AS
BEGIN
  DELETE FROM MYDBAT_PERSON_group
  WHERE groupid=group_id
  AND userid=user_id;
END DELETE_USERGROUP;

create or replace PROCEDURE DELETE_USERROLE
(user_id MYDBAT_PERSON_ROLE.USERID%TYPE,
 role_id MYDBAT_PERSON_ROLE.ROLEID%TYPE) AS
BEGIN
  DELETE FROM MYDBAT_PERSON_ROLE
  WHERE userid=user_id and roleid=role_id;
END DELETE_USERROLE;
```

Sample Delete Script

This script is invoked during revocation of an account from Oracle Access Governance. Here we are deleting the data user relationship tables, MYDBAT_PERSON_ROLE and MYDBAT_PERSON_GROUP, as well as data from the MYDBAT_PERSON table

Delete Script

```
import java.sql.PreparedStatement;
import org.identityconnectors.framework.common.objects.*;

//Get the UID from the input map 'attributes'
String uid = attributes.get("__UID__").getValue().get(0);
trace.info("[Delete-Groovy] Deleting user:: "+ uid);

try {
    //Delete data from child tables and then, main table
    //Delete user roles
    st = conn.prepareStatement("DELETE FROM MYDBAT_PERSON_ROLE WHERE
USERID=?");
    st.setString(1, uid);
    st.executeUpdate();
    st.close();

    //Delete user groups
    st = conn.prepareStatement("DELETE FROM MYDBAT_PERSON_GROUP WHERE
USERID=?");
    st.setString(1, uid);
    st.executeUpdate();
    st.close();

    //Delete user account
    st = conn.prepareStatement("DELETE FROM MYDBAT_PERSON WHERE USERID=?");
    st.setString(1, uid);
    st.executeUpdate();
```

```
} finally {
    if (st != null)
        st.close(); };
trace.info("Deleted user:: "+ uid);
```

Sample Update Script

This script is invoked during provisioning operations when account is updated from Oracle Access Governance. Here we are updating the data in MYDBAT_PERSON table

Update Script

```
import org.identityconnectors.framework.common.objects.*;
import java.text.*;
import org.identityconnectors.framework.common.exceptions.*;

trace.info("[Update-Groovy] Atrributes::"+ attributes);

/** During an Update operation,AGCS  sends the UID attribute along with
updated attributes.
Get all the values of attributes **/

String id = attributes.get("__UID__")!=null?
attributes.get("__UID__").getValue().get(0):null;
String firstName=attributes.get("FIRSTNAME")!=null?
attributes.get("FIRSTNAME").getValue().get(0):null;
String lastName=attributes.get("LASTNAME")!=null?
attributes.get("LASTNAME").getValue().get(0):null;
String email=attributes.get("EMAIL")!=null?
attributes.get("EMAIL").getValue().get(0):null;
String description=attributes.get("DESCRIPTION")!=null?
attributes.get("DESCRIPTION").getValue().get(0):null;
salary=attributes.get("SALARY")!=null?
attributes.get("SALARY").getValue().get(0):null;
joindate = attributes.get("JOININGDATE")!=null?
attributes.get("JOININGDATE").getValue().get(0):null;
enableValue = attributes.get("__ENABLE__")!=null?
attributes.get("__ENABLE__").getValue().get(0):true;

//Throw exception if uid is null
if(id==null) throw new ConnectorException("UID Cannot be Null");
    stmt = null;
try {
//Create prepared statement to update the MYDBAT_PERSON table
        stmt = conn.prepareStatement("UPDATE MYDBAT_PERSON SET
FIRSTNAME=COALESCE(?, FIRSTNAME),LASTNAME =COALESCE(?, LASTNAME), EMAIL=
COALESCE(?, EMAIL),SALARY=COALESCE(?,
SALARY),JOININGDATE=COALESCE(to_date(?,'dd-Mon-yy'),
JOININGDATE),STATUS=COALESCE(?, STATUS) WHERE USERID =?");
        //Set sql input parameters
        stmt.setString(1, firstName);
        stmt.setString(2, lastName);
        stmt.setString(3, email);
        stmt.setBigDecimal(4, new BigDecimal(salary));
        dateStr = null;
```

```
        //Convert the joindate into oracle date format
        if( joindate != null) {
            Date date=new Date(joindate);
            DateFormat targetFormat = new SimpleDateFormat("dd-MMM-yy");
            dateStr = targetFormat.format(date);
        }
        stmt.setString(5,dateStr);
        if(enableValue)
            stmt.setString(6,"Enabled");
        else
            stmt.setString(6,"Disabled");
        stmt.setString(7, id);
        stmt.executeUpdate();
} finally {
    if (stmt != null)
        stmt.close();
};
trace.info("[Update] Updated user::"+ id);
return new Uid(id);
```

Custom Scripting for Overview

When you provision accounts from Oracle Access Governance using the Database Application Tables (Oracle) integration, operations such as create, update, and delete are implemented using the default supplied code. On occasions where you wish to modify the default supplied operations, you can optionally provide your own custom scripts which implement your own specific provisioning operation requirements. This step is completely optional, you do not have to create custom scripts if the default operations provide you with what you need. You can add custom scripts to any operations supported. If you choose custom scripts, you only need to add them where you require the default operation to be modified, you can have a combination of custom and default scripts for the operations supported, though you can only have one or the other option for each specific operation. For example, the *create* operation might be implemented with a custom script that adds some functionality specific to your organization, while the *delete* operation is unchanged and uses default functionality.

Once you have implemented and configured your Database Application Tables (Oracle) to use a custom script then that script will be used when you next perform a provisioning or data load operation.

> **Note:**
>
> Any custom script must be implemented using Groovy format. Other scripting formats are not supported.

When you create a Database Application Tables (Oracle) orchestrated system you can identify scripts to be run for a number of provisioning operations on the database application containing account data. These operations are:

- Create
- Update

- Delete
- Dataload
- Add relationship data
- Remove relationship data

These scripts should be located on the agent host, in the install directory of the agent, for example, `/app/<custom script>` . You configure the agent with the location of the scripts in the integration settings for your orchestrated system. You should ensure that the operating system user running the agent has read/write permissions for any custom scripts.

When you perform a provisioning task, your script will be run as a replacement to the standard processing associated with the task. The script must handle the default provisioning task such as create or update, and can also have custom tasks above and beyond the default provisioning process, such as:

- Perform custom table updates
- Custom auditing
- Send custom notifications

This means that you have two options for provisioning processing using the Database Application Tables (Oracle) integration:

1. Use the default logic provided with the Database Application Tables (Oracle) connector
2. Use the custom logic implemented in scripts

Custom scripts are only used when configured in your orchestrated system. So, if you have specified a create script when creating your orchestrated system, but no script for update, then the custom script will be used for the create provisioning task, while the update task will be implemented using the default connector processing.

You should also note that all custom script types are supported for an orchestrated system configured for managed system mode. The only script type supported for authoritative source mode is the Dataload type, which is supported for both modes.

Sample Database Schema

The samples provided in the following sections are based on the database tables described in this section.

MYDBAT_PERSON

```
USE {dataBase};
CREATE TABLE MYDBAT_PERSON
  (USERID INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
   USERNAME VARCHAR(50) NOT NULL,
   FIRSTNAME VARCHAR(50),
   LASTNAME VARCHAR(50),
   EMAIL VARCHAR(50) NOT NULL,
   COUNTRYCODE VARCHAR(20),
   DESCRIPTION VARCHAR(MAX),
   SALARY MONEY,
   JOININGDATE DATE,
   STATUS VARCHAR(20) NOT NULL,
   PASSWORD VARCHAR(MAX));
```

MYDBAT_GROUPS

```
USE {dataBase};
   CREATE TABLE MYDBAT_GROUPS
   (GROUPID VARCHAR(50) NOT NULL PRIMARY KEY,
   GROUPNAME VARCHAR(50) NOT NULL);
```

MYDBAT_ROLES

```
USE {dataBase};
   CREATE TABLE MYDBAT_ROLES
   (ROLEID varchar(50) NOT NULL PRIMARY KEY,
   ROLENAME varchar(50) NOT NULL);
```

MYDBAT_PERSON_GROUP

```
USE {dataBase};
   CREATE TABLE MYDBAT_PERSON_GROUP
   (USERID INT NOT NULL,
   GROUPID VARCHAR(50) NOT NULL,
   CONSTRAINT MYDBAT_PERSON_GROUP_PK PRIMARY KEY (USERID, GROUPID),
   CONSTRAINT MYDBAT_PERSON_FK1 FOREIGN KEY (USERID)
     REFERENCES MYDBAT_PERSON (USERID),
   CONSTRAINT MYDBAT_GROUPS_FK1 FOREIGN KEY (GROUPID)
     REFERENCES MYDBAT_GROUPS (GROUPID));
```

MYDBAT_PERSON_ROLE

```
USE {dataBase};
   CREATE TABLE MYDBAT_PERSON_ROLE
   (USERID INT NOT NULL,
   ROLEID VARCHAR(50) NOT NULL,
   FROMDATE DATE,
   TODATE DATE,
   CONSTRAINT MYDBAT_PERSON_ROLE_PK PRIMARY KEY (USERID, ROLEID),
   CONSTRAINT MYDBAT_PERSON_FK2 FOREIGN KEY (USERID)
     REFERENCES MYDBAT_PERSON (USERID),
   CONSTRAINT MYDBAT_ROLES_FK1 FOREIGN KEY (ROLEID)
     REFERENCES MYDBAT_ROLES (ROLEID));
```

MYDBAT_COUNTRY

```
USE {dataBase};
   CREATE TABLE MYDBAT_COUNTRY
   (COUNTRYCODE VARCHAR(20) NOT NULL PRIMARY KEY,
   COUNTRYNAME VARCHAR(200) NOT NULL);
```

> **Note:**
>
> Child tables such as `mydbat_roles`, `mydbat_groups`, and `mydbat_country` should have a primary key constraint defined. If no primary key is defined for child tables then your validate operation will fail and you will see an error **Key for table <tablename> are not defined**.

Groovy Script Arguments

The following arguments can be used in your Groovy scripts:

Table 5-20 Script Arguments

Argument	Description
connector	The Database Application Tables connector object.

timing	When the Groovy script is called. The timing attribute also explains the type of operation being performed. For example, if it is a search operation, then the object class being searched is also returned.
	The following is the format of the timing argument for lookup field synchronization:
	`executeQuery:OBJECT_CLASS`
	In this format `OBJECT_CLASS` is replaced with the type of object being reconciled.
	For example, for a lookup field synchronization scheduled job that contains the object type *Role*, the value of the timing argument will be as follows:
	`executeQuery:Role`

attributes	All attributes.
trace	Logger as a script trace bridge to the application
where	String where condition for execute query, or null.
handler	resultSetHandler or SyncResultsHandler for the connector objects produced by the execute query, sync operation or null return.
quoting	The type of table name quoting to be used in SQL. The default value is an empty string. The value of this argument is obtained from the integration settings.
nativeTimestamps	Specifies whether the script retrieves the timestamp data of the columns as java.sql.Timestamp type from the database table. This information is obtained from the integration settings.

Table 5-20 (Cont.) Script Arguments

Argument	Description
allNative	Specifies whether the script must retrieve the data type of the columns in a native format from the database table. The value of this argument is obtained from the integration settings. The value of this argument specifies whether the script must throw exceptions when a zero (0x00) error code is encountered.
enableEmptyString	Specifies whether support for writing an empty string instead of a NULL value must be enabled. The value of this argument is obtained from the integration settings.
filterString	String filter condition for execute query, or null.
filterParams	List of filter parameters. Each parameter is present in the COLUMN_NAME:VALUE format. For example, FIRSTNAME:test.
syncattribute	Name of the database column configured for incremental reconciliation. This argument is available in the sync script, which is called during an incremental reconciliation run.
synctoken	Value of the sync attribute. This argument is available in the sync script.

Sample Dataload Script

The data load script reads the data from all the tables for all the defined entities. In this scenario, the term data load refers to the full data load and the lookup data load.

This sample script reads user data from the MYDBAT_PERSON table, and the users' relationship data from the MYDABAT_PERSON_ROLE and MYDBAT_PERSON_GROUP tables. Entitlements data is read from the MYDBAT_GROUPS table, and lookup data is read from the MYDBAT_COUNTRY table. It also has support for a basic filter search on MYDBAT_PERSON table. All these data reads are done using stored procedures.

Dataload Script

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.ResultSet;
import java.math.*;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.Map;
import org.identityconnectors.framework.common.objects.*;
import java.lang.reflect.*;
import org.identityconnectors.common.security.GuardedString;
import java.text.*;

String ocName ;
var df = new SimpleDateFormat("yyyy-MM-dd");
var targetFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss z");
if( timing != "") {
```

```
    trace.info("[Execute Query] timing attribute value: "+ timing);
    ocName = timing.split(":")[1]
}

trace.info("[Execute Query] for objectClass: "+ ocName);
switch (ocName) {
    //Lookup
    case "MYDBAT_COUNTRY":
        CallableStatement callableStatement = null;
        ResultSet resultSet = null;
        try {
            callableStatement = conn.prepareCall("{call GET_COUNTRIES}");
            resultSet = (ResultSet) callableStatement.executeQuery();

            while (resultSet.next()) {
                var cob = new ConnectorObjectBuilder();
                cob.setObjectClass(new ObjectClass("MYDBAT_COUNTRY"));
                cob.addAttribute(AttributeBuilder.build(Uid.NAME,
resultSet.getString(1)));
                cob.addAttribute(AttributeBuilder.build(Name.NAME,
resultSet.getString(2)));

                if(!handler.handle(cob.build())) return;
            }
        } finally {
            if(resultSet != null)
                resultSet.close();
            if(callableStatement != null)
                callableStatement.close();
        }
        break;

    //Entitlement
    case "MYDBAT_GROUPS":
        CallableStatement callableStatement = null;
        ResultSet resultSet = null;
        try {
            callableStatement = conn.prepareCall("{call GET_GROUPS}");
            resultSet = (ResultSet) callableStatement.executeQuery();

            while (resultSet.next()) {
                var cob = new ConnectorObjectBuilder();
                cob.setObjectClass(new ObjectClass("MYDBAT_GROUPS"));
                cob.addAttribute(AttributeBuilder.build(Uid.NAME,
resultSet.getString(1)));
                cob.addAttribute(AttributeBuilder.build(Name.NAME,
resultSet.getString(2)));

                if(!handler.handle(cob.build())) return;
            }
        } finally {
            if(resultSet != null)
                resultSet.close();
            if(callableStatement != null)
                callableStatement.close();
        }
```

```
        break;

    //Entitlement
    case "DBAT_ROLES":
        CallableStatement callableStatement = null;
        ResultSet resultSet = null;
        try {
            callableStatement = conn.prepareCall("{call GET_ROLES}");
            resultSet = (ResultSet) callableStatement.executeQuery();

            while (resultSet.next()) {
                var cob = new ConnectorObjectBuilder();
                cob.setObjectClass(new ObjectClass("MYDBAT_ROLES"));
                cob.addAttribute(AttributeBuilder.build(Uid.NAME,
resultSet.getString(1)));
                cob.addAttribute(AttributeBuilder.build(Name.NAME,
resultSet.getString(2)));

                if(!handler.handle(cob.build())) return;
            }
        } finally {
            if(resultSet != null)
                resultSet.close();
            if(callableStatement != null)
                callableStatement.close();
        }
        break;

    case "ACCOUNT":
    case "TARGETACCOUNT":
        CallableStatement parentCallableStatement = null;
        ResultSet parentResultSet = null;
        try {
            if (filterString != "") {
                trace.info("[Execute Query] Performing Recon with Filter.
Filter is:: "+ filterString +" And Filer Params are:: "+ filterParams);
                //[Execute Query] Performing Recon with Filter. Filter
is::MYDBAT_PERSON.USERID = ? And Filer Params are::Params are::
[MYDBAT_PERSON.USERID:31]
                parentCallableStatement = conn.prepareCall("{call
GET_PERSON_BY_USERID(?)}");
                parentCallableStatement.setString(1,
filterParams.get(0).split(":")[1]);
            } else {
                trace.info("[Execute Query] Performing Full Recon.");
                parentCallableStatement = conn.prepareCall("{call
GET_PERSONS}");
            }
            parentResultSet = (ResultSet)
parentCallableStatement.executeQuery();
            while (parentResultSet.next()) {
                var cob = new ConnectorObjectBuilder();
                cob.setObjectClass(ObjectClass.ACCOUNT);
                cob.addAttribute(AttributeBuilder.build(Uid.NAME,
parentResultSet.getString(1)));
                cob.addAttribute(AttributeBuilder.build("FIRSTNAME",
```

```
parentResultSet.getString(2)));
                cob.addAttribute(AttributeBuilder.build("LASTNAME",
parentResultSet.getString(3)));
                cob.addAttribute(AttributeBuilder.build("EMAIL",
parentResultSet.getString(4)));
                cob.addAttribute(AttributeBuilder.build("DESCRIPTION",
parentResultSet.getString(5)));
                cob.addAttribute(AttributeBuilder.build("SALARY",
parentResultSet.getDouble(6)));
                var joiningDbDate = parentResultSet.getDate(7);
                if( joiningDbDate != null ) {
                    var date = df.parse(joiningDbDate.toString());
                    var joinDateStr = targetFormat.format(date);
                    var joinDate = date.getTime();
                    trace.info("date : "+ date +" ---- joinDate : "+
joinDate);
                    trace.info("Setting joinDate: "+ joinDate);
                    cob.addAttribute(AttributeBuilder.build("JOININGDATE",
joinDate));
                }

cob.addAttribute(AttributeBuilder.build(OperationalAttributes.ENABLE_NAME,
"ACTIVE".equalsIgnoreCase(parentResultSet.getString(8))));
                cob.addAttribute(AttributeBuilder.build("COUNTRYCODE",
parentResultSet.getString(9)));
                cob.addAttribute(AttributeBuilder.build(Name.NAME,
parentResultSet.getString(10)));

                if (ocName.equals("TARGETACCOUNT")) {
                    CallableStatement callableStatement = null;
                    ResultSet resultSet = null;
                    try {
                        //Person role
                        callableStatement = conn.prepareCall("{call
GET_PERSON_ROLE(?)}");
                        callableStatement.setString(1,
parentResultSet.getString(1));
                        resultSet = (ResultSet)
callableStatement.executeQuery();

                        var eoList = new ArrayList<EmbeddedObject>();

                        while (resultSet.next()) {
                            var roleEA = new EmbeddedObjectBuilder();
                            roleEA.setObjectClass(new
ObjectClass("MYDBAT_ROLES"));

roleEA.addAttribute(AttributeBuilder.build("ROLEID", resultSet.getString(2)));

                            var fromDbDate = resultSet.getDate(3);
                            if( fromDbDate != null ) {
                                var date = df.parse(fromDbDate.toString());
                                var fromDateStr = targetFormat.format(date);
                                var fromDate = date.getTime();
                                trace.info("Setting roles fromDate : "+
fromDate);
```

```
roleEA.addAttribute(AttributeBuilder.build("FROMDATE", fromDate));
                            }

                            var toDbDate = resultSet.getDate(4);
                            if( toDbDate != null ) {
                                var date = df.parse(toDbDate.toString());
                                var toDateStr = targetFormat.format(date);
                                var toDate = date.getTime();
                                trace.info("Setting roles toDate: "+ toDate);

roleEA.addAttribute(AttributeBuilder.build("TODATE", toDate));
                            }
                            eoList.add(roleEA.build());
                        }

                        var roleEm = eoList.toArray(new
EmbeddedObject[eoList.size()]);

cob.addAttribute(AttributeBuilder.build("MYDBAT_PERSON_ROLE", (Object[])
roleEm));
                    } finally {
                        if(resultSet != null)
                            resultSet.close();
                        if(callableStatement != null)
                            callableStatement.close();
                    }

                    try {
                        //Person group
                        callableStatement = conn.prepareCall("{call
GET_PERSON_GROUP(?)}");
                        callableStatement.setString(1,
parentResultSet.getString(1));
                        resultSet = (ResultSet)
callableStatement.executeQuery();

                        var geoList = new ArrayList<EmbeddedObject>();

                        while (resultSet.next()) {
                            var groupEA = new EmbeddedObjectBuilder();
                            groupEA.setObjectClass(new
ObjectClass("MYDBAT_GROUPS"));

groupEA.addAttribute(AttributeBuilder.build("GROUPID",
resultSet.getString(2)));
                            geoList.add(groupEA.build());
                        }

                        var groupEm = geoList.toArray(new
EmbeddedObject[geoList.size()]);

cob.addAttribute(AttributeBuilder.build("DBAT_PERSON_GROUP", (Object[])
groupEm));
                    } finally {
                        if( resultSet != null )
```

```
                        resultSet.close();
                    if( callableStatement != null )
                        callableStatement.close();
                }
            }

            if(!handler.handle(cob.build())) return;
        }
    } finally {
        if( parentResultSet != null )
            parentResultSet.close();
        if( parentCallableStatement != null )
            parentCallableStatement.close();
    }
    break;
}
```

Stored Procedure: Load Users

```
USE {dataBase};

CREATE OR ALTER PROCEDURE GET_PERSONS
AS
BEGIN
    SELECT USERID,
           FIRSTNAME,
           LASTNAME,
           EMAIL,
           DESCRIPTION,
           SALARY,
           JOININGDATE,
           STATUS,
           COUNTRYCODE,
           USERNAME
    FROM MYDBAT_PERSON
END;
```

Stored Procedure: Filtered User Search

```
USE {dataBase};

CREATE OR ALTER PROCEDURE GET_PERSON_BY_USERID
@user_id INT
AS
BEGIN
   SELECT USERID,
          FIRSTNAME,
          LASTNAME,
          EMAIL,
          DESCRIPTION,
          SALARY,
          JOININGDATE,
          STATUS,
          COUNTRYCODE,
          USERNAME
```

```
   FROM MYDBAT_PERSON
   WHERE USERID = @user_id;
END;
```

This is a very basic example of filter search with only one filter condition, for example, *MYDBAT_PERSON.USERID:21*. This is used specifically for writeBack processing after the create operation

Stored Procedure: Get Roles

```
USE {dataBase};

CREATE OR ALTER PROCEDURE GET_ROLES
AS
BEGIN
   SELECT ROLEID,
          ROLENAME
   FROM MYDBAT_ROLES;
END;
```

Stored Procedure: Get User Roles

```
USE {dataBase};

CREATE OR ALTER PROCEDURE GET_PERSON_ROLE
@user_id INT
AS
BEGIN
   SELECT USERID,
          ROLEID,
          FROMDATE,
          TODATE
   FROM MYDBAT_PERSON_ROLE WHERE USERID = @user_id
 END;
```

Stored Procedure: Get Groups

```
USE {dataBase};

CREATE OR ALTER PROCEDURE GET_GROUPS
AS
BEGIN
   SELECT GROUPID,
          GROUPNAME
   FROM MYDBAT_GROUPS;
 END;
```

Stored Procedure: Get User Groups

```
USE {dataBase};

CREATE OR ALTER PROCEDURE GET_PERSON_GROUP
@user_id INT
```

```
AS
BEGIN
   SELECT USERID,
          GROUPID
   FROM MYDBAT_PERSON_GROUP where USERID = @user_id;
END;
```

Stored Procedure: Get Lookups (Country)

```
use {dataBase};

CREATE OR ALTER PROCEDURE GET_COUNTRIES
AS
BEGIN
   SELECT COUNTRYCODE,
          COUNTRYNAME
   FROM MYDBAT_COUNTRY;
 END;
```

Sample Create Script

This script is invoked during provisioning of a new account from Oracle Access Governance. Here we are inserting data into the MYDBAT_PERSON table.

Create Script

```
import java.sql.CallableStatement;
import java.sql.Types;
import java.util.Date.*;
import org.identityconnectors.common.security.GuardedString;
import org.identityconnectors.framework.common.objects.*;
import java.text.*;

CallableStatement callableStatement = null;
String uid = null;
try {
    trace.info("[Create-Groovy] Attributes:: " + attributes);

    callableStatement = conn.prepareCall("{call
ADD_PERSON(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}");
    callableStatement.setString(1, attributes.get(Name.NAME) != null ?
attributes.get(Name.NAME).getValue().get(0) : null);
    callableStatement.setString(2, attributes.get("FIRSTNAME") != null ?
attributes.get("FIRSTNAME").getValue().get(0) : null);
    callableStatement.setString(3, attributes.get("LASTNAME") != null ?
attributes.get("LASTNAME").getValue().get(0) : null);
    callableStatement.setString(4, attributes.get("EMAIL") != null ?
attributes.get("EMAIL").getValue().get(0) : null);
    callableStatement.setString(5, attributes.get("COUNTRYCODE") != null ?
attributes.get("COUNTRYCODE").getValue().get(0) : null);
    callableStatement.setString(6, attributes.get("DESCRIPTION") != null ?
attributes.get("DESCRIPTION").getValue().get(0) : null);
    callableStatement.setString(7,
(attributes.get(OperationalAttributes.ENABLE_NAME) != null ?
attributes.get(OperationalAttributes.ENABLE_NAME).getValue().get(0) : true) ?
```

```
"ACTIVE" : "INACTIVE");

    var joiningdate = attributes.get("JOININGDATE") != null ?
attributes.get("JOININGDATE").getValue().get(0) : null;
    callableStatement.setString(8, (joiningdate != null && joiningdate !=
0) ? new SimpleDateFormat("yyyy-MM-dd").format(new Date(joiningdate)) :
null);

    var salary = attributes.get("SALARY") != null ?
attributes.get("SALARY").getValue().get(0) : null;
    if(salary != null)
        callableStatement.setDouble(9, salary);
    else
        callableStatement.setNull(9, Types.DOUBLE)

    var password = attributes.get(OperationalAttributes.PASSWORD_NAME) !=
null ?
attributes.get(OperationalAttributes.PASSWORD_NAME).getValue().get(0) : null;
    if (password != null) {
        password.access(new GuardedString.Accessor() {
            public void access(char[] clearChars)
{ callableStatement.setString(10, new String(clearChars));}
        });
    } else
        callableStatement.setString(10, null);

    callableStatement.registerOutParameter(11, Types.NVARCHAR);
    callableStatement.execute();

    uid = callableStatement.getString(11);
} finally {
    if (callableStatement != null)
        callableStatement.close();
}
trace.info("[Create] Created User:: " +uid);
return new Uid(uid);
```

Sample Add Child Script

This script is invoked during provisioning of entitlements/permissions to users from Oracle Access Governance. Here we are inserting data into the MYDBAT_PERSON_GROUP and MYDBAT_PERSON_ROLE tables.

Add Child Script

```
import java.sql.PreparedStatement;
import org.identityconnectors.framework.common.objects.*;
import java.text.*;

trace.info("[addMultiValuedAttributeScript-Groovy] Adding Child data:: " +
attributes);
String userId = attributes.get(Uid.NAME).getValue().get(0);
PreparedStatement childst = null;
Attribute dbatChild = null;
try {
```

```
    dbatChild = attributes.get("MYDBAT_PERSON_GROUP");
    if (dbatChild != null) {
        var childDataEOSet = dbatChild.getValue();
        childst = conn.prepareStatement("INSERT INTO MYDBAT_PERSON_GROUP
VALUES (?, ?)");

        if (childDataEOSet != null) {
            trace.info("[addMultiValuedAttributeScript] Adding Group data.");
            for ( iterator = childDataEOSet.iterator(); iterator.hasNext(); )
{
                eo = iterator.next();
                attrsSet = eo.getAttributes();
                grpAttr = AttributeUtil.find("GROUPID", attrsSet);
                if (grpAttr != null) {
                    childst.setString(1, userId);
                    childst.setString(2, grpAttr.getValue().get(0));
                    childst.executeUpdate();
                    childst.clearParameters();
                }
            };
        }
    }
} finally {
    if (childst != null)
        childst.close();
};

try {
    dbatChild = attributes.get("MYDBAT_PERSON_ROLE");
    if (dbatChild != null) {
        var childDataEOSet = dbatChild.getValue();
        childst = conn.prepareStatement("INSERT INTO MYDBAT_PERSON_ROLE
VALUES (?, ?, ?, ?)");

        if (childDataEOSet != null) {
            trace.info("[addMultiValuedAttributeScript] Adding Role data.");
            for ( iterator = childDataEOSet.iterator(); iterator.hasNext(); )
{
                eo = iterator.next();
                attrsSet = eo.getAttributes();
                roleattr = AttributeUtil.find("ROLEID", attrsSet);

                if (roleattr != null) {
                    childst.setString(1, userId);
                    childst.setString(2, roleattr.getValue().get(0));
                    childst.setString(3, AttributeUtil.find("FROMDATE",
attrsSet) != null ? new SimpleDateFormat("yyyy-MM-dd").format(new
Date(AttributeUtil.find("FROMDATE", attrsSet).getValue().get(0))) : null);
                    childst.setString(4, AttributeUtil.find("TODATE",
attrsSet) != null ? new SimpleDateFormat("yyyy-MM-dd").format(new
Date(AttributeUtil.find("TODATE", attrsSet).getValue().get(0))) : null);

                    childst.executeUpdate();
                    childst.clearParameters();
                }
            };
```

```
        }
    }
} finally {
    if (childst != null)
        childst.close();
};
```

Sample Remove Child Script

This script is invoked during deprovisioning of entitlements/permissions from users from Oracle Access Governance. Here we are removing data from MYDBAT_PERSON_GROUP and MYDBAT_PERSON_ROLE tables using stored procedures.

Remove Child Script

```
import java.sql.CallableStatement;
import org.identityconnectors.framework.common.objects.*;

trace.info("[removeMultiValuedAttributeScript] Removing Child data:: "+
attributes);

var uid = attributes.get(Uid.NAME).getValue().get(0);
CallableStatement callableStatement = null;
Attribute dbatChild = null;
try {
    dbatChild = attributes.get("MYDBAT_PERSON_GROUP");
    if (dbatChild != null) {
        var childDataEOSet = dbatChild.getValue();
        //Delete child data using stored procedure
        callableStatement = conn.prepareCall("{call
DELETE_PERSON_GROUP(?, ?)}");
        if(childDataEOSet != null) {
            trace.info("[removeMultiValuedAttributeScript] Removing Group
data.");
            //Iterate through child data and delete
            for( iterator = childDataEOSet.iterator(); iterator.hasNext(); ) {
                eo = iterator.next();
                grpattr = AttributeUtil.find("GROUPID", eo.getAttributes());
                if (grpattr != null) {
                    callableStatement.setString(1, uid);
                    callableStatement.setString(2, grpattr.getValue().get(0));
                    callableStatement.executeUpdate();
                    trace.info("[removeMultiValuedAttributeScript] Deleted
Group:: "+ grpattr);
                }
            };
        }
    }
} finally {
    if (callableStatement != null)
        callableStatement.close();
};

try {
    dbatChild = attributes.get("MYDBAT_PERSON_ROLE");
```

```
    if (dbatChild != null) {
        var childDataEOSet = dbatChild.getValue();
        callableStatement = conn.prepareCall("{call
DELETE_PERSON_ROLE(?, ?)}");
        if(childDataEOSet != null) {
            trace.info("[removeMultiValuedAttributeScript] Removing Role
data.");
            for ( iterator = childDataEOSet.iterator(); iterator.hasNext(); )
{
                eo = iterator.next();
                roleattr = AttributeUtil.find("ROLEID", eo.getAttributes());
                if(roleattr != null) {
                    callableStatement.setString(1, uid);
                    callableStatement.setString(2,
roleattr.getValue().get(0));
                    callableStatement.executeUpdate();
                    trace.info("[removeMultiValuedAttributeScript] Deleted
Role:: "+ roleattr);
                }
            };
        }
    }
} finally {
    if (callableStatement != null)
        callableStatement.close();
};
```

Stored Procedure: Remove Child

```
USE {dataBase};

CREATE OR ALTER PROCEDURE DELETE_PERSON_GROUP
 @user_id INT, @group_id nvarchar(50)
 AS
 BEGIN
    DELETE from MYDBAT_PERSON_GROUP where USERID = @user_id AND GROUPID =
@group_id
 END;

USE {dataBase};

CREATE OR ALTER PROCEDURE DELETE_PERSON_ROLE
 @user_id INT, @role_id nvarchar(50)
 AS
 BEGIN
   DELETE FROM MYDBAT_PERSON_ROLE where USERID = @user_id AND ROLEID =
@role_id
 END;
```

Sample Delete Script

This script is invoked during revocation of an account from Oracle Access Governance. Here we are deleting the data user relationship tables, MYDBAT_PERSON_ROLE and MYDBAT_PERSON_GROUP, as well as data from the MYDBAT_PERSON table

Delete Script

```
import java.sql.CallableStatement;
import org.identityconnectors.framework.common.objects.*;

var uid = attributes.get(Uid.NAME).getValue().get(0);
CallableStatement callableStatement = null;
try {
    trace.info("[Delete-Groovy] Deleting user:: " + uid);

    callableStatement = conn.prepareCall("{call DELETE_PERSON(?)}");
    callableStatement.setString(1, uid);
    callableStatement.execute();
} finally {
    if (callableStatement != null)
        callableStatement.close();
};
trace.info("Deleted user:: " + uid);
```

Stored Procedure: Delete

```
USE {dataBase};

CREATE OR ALTER PROCEDURE DELETE_PERSON
 @user_id INT
 AS
 BEGIN
   DELETE FROM MYDBAT_PERSON_ROLE where USERID = @user_id;
   DELETE FROM MYDBAT_PERSON_GROUP where USERID = @user_id;
   DELETE FROM MYDBAT_PERSON WHERE USERID = @user_id;
 END;
```

Sample Update Script

This script is invoked during provisioning operations when account is updated from Oracle Access Governance. Here we are updating the data in MYDBAT_PERSON table

Update Script

```
import java.sql.PreparedStatement;
import java.sql.Types;
import org.identityconnectors.framework.common.objects.*;
import java.text.*;
import org.identityconnectors.framework.common.exceptions.*;
import org.identityconnectors.common.security.GuardedString;

trace.info("[Update-Groovy] Atrributes:: " + attributes);
```

```
PreparedStatement stmt = null;
String userId = null;
try {
    userId = attributes.get(Uid.NAME) != null ?
attributes.get(Uid.NAME).getValue().get(0) : null;

    if (userId == null)
        throw new ConnectorException("UID Cannot be Null");

    stmt = conn.prepareStatement("UPDATE MYDBAT_PERSON SET FIRSTNAME =
COALESCE(?, FIRSTNAME), LASTNAME = COALESCE(?, LASTNAME), EMAIL = COALESCE(?,
EMAIL), COUNTRYCODE = COALESCE(?, COUNTRYCODE), DESCRIPTION = COALESCE(?,
DESCRIPTION), STATUS = COALESCE(?, STATUS), JOININGDATE = COALESCE(?,
JOININGDATE), SALARY = COALESCE(?, SALARY), PASSWORD = COALESCE(?, PASSWORD)
WHERE USERID = ?");
    stmt.setString(1, attributes.get("FIRSTNAME") != null ?
attributes.get("FIRSTNAME").getValue().get(0) : null);
    stmt.setString(2, attributes.get("LASTNAME") != null ?
attributes.get("LASTNAME").getValue().get(0) : null);
    stmt.setString(3, attributes.get("EMAIL") != null ?
attributes.get("EMAIL").getValue().get(0) : null);
    stmt.setString(4, attributes.get("COUNTRYCODE") != null ?
attributes.get("COUNTRYCODE").getValue().get(0) : null);
    stmt.setString(5, attributes.get("DESCRIPTION") != null ?
attributes.get("DESCRIPTION").getValue().get(0) : null);
    stmt.setString(6, (attributes.get(OperationalAttributes.ENABLE_NAME) !=
null ? attributes.get(OperationalAttributes.ENABLE_NAME).getValue().get(0) :
true) ? "ACTIVE" : "INACTIVE");

    var joiningdate = attributes.get("JOININGDATE") != null ?
attributes.get("JOININGDATE").getValue().get(0) : null;
    stmt.setString(7, joiningdate != null ? new SimpleDateFormat("yyyy-MM-
dd").format(new Date(joiningdate)) : null)

    var salary = attributes.get("SALARY") != null ?
attributes.get("SALARY").getValue().get(0) : null;
    if(salary != null)
        stmt.setDouble(8, salary);
    else
        stmt.setNull(8, Types.DOUBLE)

    var password = attributes.get(OperationalAttributes.PASSWORD_NAME) !=
null ?
attributes.get(OperationalAttributes.PASSWORD_NAME).getValue().get(0) : null;
    if (password != null) {
        password.access(new GuardedString.Accessor() {
            public void access(char[] clearChars) { stmt.setString(9, new
String(clearChars));}
        });
    } else
        stmt.setString(9, null);
    stmt.setString(10, userId);

    stmt.executeUpdate();
} finally {
```

```
    if (stmt != null)
        stmt.close();
};
trace.info("[Update] Updated user:: " + userId);
return new Uid(userId);
```

Custom Scripting for Overview

When you provision accounts from Oracle Access Governance using the Database Application Tables (Oracle) integration, operations such as create, update, and delete are implemented using the default supplied code. On occasions where you wish to modify the default supplied operations, you can optionally provide your own custom scripts which implement your own specific provisioning operation requirements. This step is completely optional, you do not have to create custom scripts if the default operations provide you with what you need. You can add custom scripts to any operations supported. If you choose custom scripts, you only need to add them where you require the default operation to be modified, you can have a combination of custom and default scripts for the operations supported, though you can only have one or the other option for each specific operation. For example, the *create* operation might be implemented with a custom script that adds some functionality specific to your organization, while the *delete* operation is unchanged and uses default functionality.

Once you have implemented and configured your Database Application Tables (Oracle) to use a custom script then that script will be used when you next perform a provisioning or data load operation.

> **Note:**
>
> Any custom script must be implemented using Groovy format. Other scripting formats are not supported.

When you create a Database Application Tables (Oracle) orchestrated system you can identify scripts to be run for a number of provisioning operations on the database application containing account data. These operations are:

- Create
- Update
- Delete
- Dataload
- Add relationship data
- Remove relationship data

These scripts should be located on the agent host, in the install directory of the agent, for example, `/app/<custom script>` . You configure the agent with the location of the scripts in the integration settings for your orchestrated system. You should ensure that the operating system user running the agent has read/write permissions for any custom scripts.

When you perform a provisioning task, your script will be run as a replacement to the standard processing associated with the task. The script must handle the default provisioning task such

as create or update, and can also have custom tasks above and beyond the default provisioning process, such as:

- Perform custom table updates
- Custom auditing
- Send custom notifications

This means that you have two options for provisioning processing using the Database Application Tables (Oracle) integration:

1. Use the default logic provided with the Database Application Tables (Oracle) connector
2. Use the custom logic implemented in scripts

Custom scripts are only used when configured in your orchestrated system. So, if you have specified a create script when creating your orchestrated system, but no script for update, then the custom script will be used for the create provisioning task, while the update task will be implemented using the default connector processing.

You should also note that all custom script types are supported for an orchestrated system configured for managed system mode. The only script type supported for authoritative source mode is the Dataload type, which is supported for both modes.

Sample Database Schema

The samples provided in the following sections are based on the database tables described in this section.

MYDBAT_PERSON

```
USE {dataBase};

CREATE TABLE MYDBAT_PERSON (
    USERID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    USERNAME VARCHAR(50) NOT NULL,
    FIRSTNAME VARCHAR(50),
    LASTNAME VARCHAR(50),
    EMAIL VARCHAR(50) NOT NULL,
    COUNTRYCODE VARCHAR(20),
    DESCRIPTION VARCHAR(50),
    SALARY DECIMAL(10, 2),
    JOININGDATE DATE,
    STATUS VARCHAR(50),
    LASTUPDATED TIMESTAMP(6),
    PASSWORD VARCHAR(50));
```

MYDBAT_GROUPS

```
USE {dataBase};

CREATE TABLE MYDBAT_GROUPS (
    GROUPID VARCHAR(20) NOT NULL,
    GROUPNAME VARCHAR(20) NOT NULL,
    PRIMARY KEY (GROUPID));
```

MYDBAT_ROLES

```
USE {dataBase};

CREATE TABLE MYDBAT_ROLES (
    ROLEID VARCHAR(50) NOT NULL PRIMARY KEY,
    ROLENAME VARCHAR(50) NOT NULL);
```

MYDBAT_PERSON_GROUP

```
USE {dataBase};

CREATE TABLE MYDBAT_PERSON_GROUP(
    USERID INT NOT NULL,
    GROUPID VARCHAR(20) NOT NULL,
    PRIMARY KEY (USERID,GROUPID));

ALTER TABLE DBAT_PERSON_GROUP
ADD CONSTRAINT FK_USERID
FOREIGN KEY (USERID) REFERENCES DBAT_PERSON(USERID);
```

MYDBAT_PERSON_ROLE

```
USE {dataBase};

CREATE TABLE MYDBAT_PERSON_ROLE(
    USERID INT NOT NULL,
    ROLEID VARCHAR(20) NOT NULL,
    PRIMARY KEY (USERID,ROLEID));

ALTER TABLE MYDBAT_PERSON_ROLE
ADD CONSTRAINT FK_USERIDROLE
FOREIGN KEY (USERID) REFERENCES DBAT_PERSON(USERID);
```

MYDBAT_COUNTRY

```
USE {dataBase};

CREATE TABLE MYDBAT_COUNTRY(
    COUNTRYCODE VARCHAR(20) NOT NULL,
    COUNTRYNAME VARCHAR(20) NOT NULL,
    PRIMARY KEY (COUNTRYCODE)
);
```

Note:

Child tables such as `mydbat_roles`, `mydbat_groups`, and `mydbat_country` should have a primary key constraint defined. If no primary key is defined for child tables then your validate operation will fail and you will see an error **Key for table <tablename> are not defined**.

Groovy Script Arguments

The following arguments can be used in your Groovy scripts:

Table 5-21 Script Arguments

Argument	Description
connector	The Database Application Tables connector object.

timing	When the Groovy script is called. The timing attribute also explains the type of operation being performed. For example, if it is a search operation, then the object class being searched is also returned.
	The following is the format of the timing argument for lookup field synchronization:
	`executeQuery:OBJECT_CLASS`
	In this format `OBJECT_CLASS` is replaced with the type of object being reconciled.
	For example, for a lookup field synchronization scheduled job that contains the object type *Role*, the value of the timing argument will be as follows:
	`executeQuery:Role`

attributes	All attributes.
trace	Logger as a script trace bridge to the application
where	String where condition for execute query, or null.
handler	resultSetHandler or SyncResultsHandler for the connector objects produced by the execute query, sync operation or null return.
quoting	The type of table name quoting to be used in SQL. The default value is an empty string. The value of this argument is obtained from the integration settings.
nativeTimestamps	Specifies whether the script retrieves the timestamp data of the columns as java.sql.Timestamp type from the database table. This information is obtained from the integration settings.
allNative	Specifies whether the script must retrieve the data type of the columns in a native format from the database table. The value of this argument is obtained from the integration settings. The value of this argument specifies whether the script must throw exceptions when a zero (0x00) error code is encountered.
enableEmptyString	Specifies whether support for writing an empty string instead of a NULL value must be enabled. The value of this argument is obtained from the integration settings.
filterString	String filter condition for execute query, or null.

Table 5-21 (Cont.) Script Arguments

Argument	Description
filterParams	List of filter parameters. Each parameter is present in the COLUMN_NAME:VALUE format. For example, FIRSTNAME:test.
syncattribute	Name of the database column configured for incremental reconciliation. This argument is available in the sync script, which is called during an incremental reconciliation run.
synctoken	Value of the sync attribute. This argument is available in the sync script.

Sample Dataload Script

The data load script reads the data from all the tables for all the defined entities. In this scenario, the term data load refers to the full data load and the lookup data load.

This sample script reads user data from the MYDBAT_PERSON table, and the users' relationship data from the MYDABAT_PERSON_ROLE and MYDBAT_PERSON_GROUP tables. Entitlements data is read from the MYDBAT_GROUPS table, and lookup data is read from the MYDBAT_COUNTRY table. It also has support for a basic filter search on MYDBAT_PERSON table. All these data reads are done using stored procedures.

Dataload Script

```
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.ResultSet;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import org.identityconnectors.framework.common.objects.*;
import org.identityconnectors.common.security.GuardedString;

ResultSet rs = null;
CallableStatement st = null;

try {
    String ocName = "";
    if (timing != "") {
        trace.info("[Execute Query] timing attribute value: " + timing);
        ocName = timing.split(":")[1];
    }

    trace.info("[Execute Query] for objectClass: " + ocName);

    if (ocName.equals("ACCOUNT") || ocName.equals("TARGETACCOUNT")) {
        if (filterString != "") {
            trace.info("[Execute Query] Performing Recon with Filter. Filter
is: " + filterString + " And Filter Params are: " + filterParams);
            // Example: Filter is MYDBAT_PERSON.USERID = ? And Filter Params
are [MYDBAT_PERSON.USERID:21]
            String[] filter = filterParams.get(0).split(":");
```

```
            st = conn.prepareCall("{call EXECUTE_QUERY_WITH_FILTER(?, ?)}");
            st.setString(1, filter[0]); // Column name (e.g.,
MYDBAT_PERSON.USERID)
            st.setInt(2, Integer.parseInt(filter[1])); // Value to filter
(e.g., 21)
        } else {
            trace.info("[Execute Query] Performing Full Recon.");
            st = conn.prepareCall("{call EXECUTE_QUERY_PERSON()}");
        }

        // Execute the procedure and get the ResultSet
        rs = st.executeQuery(); // No need to register OUT parameter
        SimpleDateFormat targetFormat = new SimpleDateFormat("yyyy/MM/dd
HH:mm:ss z");
        DateFormat df = new SimpleDateFormat("yyyy-MM-dd");

        while (rs.next()) {
            ConnectorObjectBuilder cob = new ConnectorObjectBuilder();
            cob.setObjectClass(ObjectClass.ACCOUNT);

            Attribute fname = AttributeBuilder.build("FIRSTNAME",
rs.getString("FIRSTNAME"));
            Attribute lname = AttributeBuilder.build("LASTNAME",
rs.getString("LASTNAME"));
            Attribute uid = AttributeBuilder.build("__UID__",
rs.getString("USERID"));
            Attribute name = AttributeBuilder.build("__NAME__",
rs.getString("USERNAME"));
            Attribute email = AttributeBuilder.build("EMAIL",
rs.getString("EMAIL"));
            Attribute description = AttributeBuilder.build("DESCRIPTION",
rs.getString("DESCRIPTION"));

            Date dbDate = rs.getDate("JOININGDATE");
            String joinDateStr = null;
            Long joinDate = null;
            if (dbDate != null) {
                java.util.Date date = df.parse(dbDate.toString());
                joinDateStr = targetFormat.format(date);
                joinDate = date.getTime();
                trace.info("date: " + date + " ---- joinDate: " + joinDate);
            }

            if (joinDate != null) {
                trace.info("Setting joinDate: " + joinDate);
                Attribute joindate = AttributeBuilder.build("JOININGDATE",
joinDate);
                cob.addAttribute(joindate);
            }

            Attribute status = AttributeBuilder.build("STATUS",
rs.getString("STATUS"));
            Attribute countryCode = AttributeBuilder.build("COUNTRYCODE",
rs.getString("COUNTRYCODE"));

            cob.addAttribute(fname);
```

```
            cob.addAttribute(lname);
            cob.addAttribute(uid);
            cob.addAttribute(name);
            cob.addAttribute(email);
            cob.addAttribute(description);
            cob.addAttribute(status);
            cob.addAttribute(countryCode);

            if (!handler.handle(cob.build())) return;
        }
    } else if (ocName.equals("DBAT_COUNTRY")) {
        trace.info("[Execute Query] for Lookup: " + ocName);
        CallableStatement countryStmt = conn.prepareCall("{call
GET_COUNTRIES()}");
        rs = countryStmt.executeQuery();

        while (rs.next()) {
            ConnectorObjectBuilder cob = new ConnectorObjectBuilder();
            cob.setObjectClass(new ObjectClass("DBAT_COUNTRY"));
            Attribute groupId = AttributeBuilder.build("__UID__",
rs.getString(1));
            Attribute groupName = AttributeBuilder.build("__NAME__",
rs.getString(2));
            cob.addAttribute(groupId);
            cob.addAttribute(groupName);
            if (!handler.handle(cob.build())) return;
        }

        rs.close();
        countryStmt.close();
    } else if (ocName.equals("DBAT_GROUPS")) {
        trace.info("[Execute Query] for Entitlement: " + ocName);
        CallableStatement groupStmt = conn.prepareCall("{call GET_GROUPS()}");
        rs = groupStmt.executeQuery();

        while (rs.next()) {
            ConnectorObjectBuilder cob = new ConnectorObjectBuilder();
            cob.setObjectClass(new ObjectClass("DBAT_GROUPS"));
            Attribute groupId = AttributeBuilder.build("__UID__",
rs.getString(1));
            Attribute groupName = AttributeBuilder.build("__NAME__",
rs.getString(2));
            cob.addAttribute(groupId);
            cob.addAttribute(groupName);
            if (!handler.handle(cob.build())) return;
        }

        rs.close();
        groupStmt.close();
    }else if (ocName.equals("DBAT_ROLES")) {
        trace.info("[Execute Query] for Entitlement: " + ocName);
        CallableStatement groupStmt = conn.prepareCall("{call GET_ROLES()}");
        rs = groupStmt.executeQuery();

        while (rs.next()) {
            ConnectorObjectBuilder cob = new ConnectorObjectBuilder();
```

```
            cob.setObjectClass(new ObjectClass("DBAT_ROLESS"));
            Attribute groupId = AttributeBuilder.build("__UID__",
rs.getString(1));
            Attribute groupName = AttributeBuilder.build("__NAME__",
rs.getString(2));
            cob.addAttribute(groupId);
            cob.addAttribute(groupName);
            if (!handler.handle(cob.build())) return;
        }

        rs.close();
        groupStmt.close();
    }
} finally {
    if (rs != null) rs.close();
    if (st != null) st.close();
}
```

Stored Procedure: Load Users

```
USE {dataBase};

DELIMITER //
CREATE PROCEDURE EXECUTE_QUERY_PERSON()
BEGIN
  SELECT USERID,
         FIRSTNAME,
         LASTNAME,
         EMAIL,
         DESCRIPTION,
         SALARY,
         JOININGDATE,
         STATUS,
         COUNTRYCODE,
         USERNAME
  FROM MYDBAT_PERSON;
END //
DELIMITER ;
```

Stored Procedure: Filtered User Search

```
USE {dataBase};

DELIMITER //
CREATE PROCEDURE EXECUTE_QUERY_WITH_FILTER(
  IN filter VARCHAR(255),
  IN filterValue int
)
BEGIN
  SET @sql_query = CONCAT(
    'SELECT USERID,
            FIRSTNAME,
            LASTNAME,
            EMAIL,
            DESCRIPTION,
```

```
            SALARY,
            JOININGDATE,
            STATUS,
            COUNTRYCODE,
            USERNAME ',
    'FROM MYDBAT_PERSON WHERE ', filter, ' = ?'
  );

  PREPARE stmt FROM @sql_query;
  SET @filter_value = filterValue;

  EXECUTE stmt USING @filter_value;

  DEALLOCATE PREPARE stmt;
END //
DELIMITER ;
```

This is a very basic example of filter search with only one filter condition, for example, *MYDBAT_PERSON.USERID:21*. This is used specifically for writeBack processing after the create operation

Stored Procedure: Get Roles

```
USE {dataBase};

DELIMITER //
CREATE PROCEDURE GET_ROLES()
BEGIN
  SELECT ROLEID,
         ROLENAME
  FROM MYDBAT_ROLES;
END //
DELIMITER ;
```

Stored Procedure: Get User Roles

```
USE {dataBase};

DELIMITER //
CREATE PROCEDURE GET_USERROLE(
  IN userin int
)
BEGIN
  SELECT USERID,
         ROLEID,
         FROMDATE,
         TODATE
  FROM MYDBAT_PERSON_ROLE
  WHERE USERID = userin;
END //
DELIMITER ;
```

Stored Procedure: Get Groups

```
USE {dataBase};

DELIMITER //
CREATE PROCEDURE GET_GROUPS()
BEGIN
  SELECT GROUPID,
         GROUPNAME
  FROM MYDBAT_GROUPS;
END //
DELIMITER ;
```

Stored Procedure: Get User Groups

```
USE {dataBase};

DELIMITER //
CREATE PROCEDURE GET_USERGROUP(
  IN userin int
)
BEGIN
  SELECT USERID,
         GROUPID
  FROM MYDBAT_PERSON_GROUP
  WHERE USERID = userin;
END //
DELIMITER ;
```

Stored Procedure: Get Lookups (Country)

```
USE {dataBase};

DELIMITER //
CREATE PROCEDURE GET_COUNTRIES()
BEGIN
  SELECT COUNTRYCODE,
         COUNTRYNAME
  FROM MYDBAT_COUNTRY;
END //
DELIMITER ;
```

Sample Create Script

This script is invoked during provisioning of a new account from Oracle Access Governance. Here we are inserting data into the MYDBAT_PERSON table.

Create Script

```
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.CallableStatement;
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
import org.identityconnectors.common.security.GuardedString;
import org.identityconnectors.framework.common.objects.*;

trace.info("[Create-Groovy] Attributes::" + attributes);

// Get all the attributes from script argument
GuardedString pass = attributes.get("__PASSWORD__") != null ?
attributes.get("__PASSWORD__").getValue().get(0) : null;
String uname = attributes.get("__NAME__") != null ?
attributes.get("__NAME__").getValue().get(0) : null;
boolean enableValue = attributes.get("__ENABLE__") != null ?
attributes.get("__ENABLE__").getValue().get(0) : true;
String email = attributes.get("EMAIL") != null ?
attributes.get("EMAIL").getValue().get(0) : null;
String first = attributes.get("FIRSTNAME") != null ?
attributes.get("FIRSTNAME").getValue().get(0) : null;
String last = attributes.get("LASTNAME") != null ?
attributes.get("LASTNAME").getValue().get(0) : null;
String desc = attributes.get("DESCRIPTION") != null ?
attributes.get("DESCRIPTION").getValue().get(0) : null;
Object salary = attributes.get("SALARY") != null ?
attributes.get("SALARY").getValue().get(0) : null; // Changed to Object
String countryCode = attributes.get("COUNTRYCODE") != null ?
attributes.get("COUNTRYCODE").getValue().get(0) : null;
Object joiningdate = attributes.get("JOININGDATE") != null ?
attributes.get("JOININGDATE").getValue().get(0) : null; // Changed to Object

// Prepare callable statement for the stored procedure
CallableStatement createStmt = null;

try {
    // Prepare the SQL statement to call the stored procedure
    createStmt = conn.prepareCall("{CALL
ADD_PERSON(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)}");

    // Set the input parameters for the stored procedure
    createStmt.setString(1, uname);    // USERNAME
    createStmt.setString(2, first);    // FIRSTNAME
    createStmt.setString(3, last);     // LASTNAME
    createStmt.setString(4, email);    // EMAIL
    createStmt.setString(5, countryCode); // COUNTRYCODE
    createStmt.setString(6, desc);     // DESCRIPTION

    // Handling SALARY: Ensure it's numeric or set as NULL
    if (salary != null && salary instanceof String) {
        try {
            createStmt.setBigDecimal(9, new BigDecimal((String) salary)); //
Parse to BigDecimal
        } catch (NumberFormatException e) {
            trace.error("[Create-Groovy] Invalid SALARY format: " + salary);
            createStmt.setNull(9, java.sql.Types.DECIMAL); // Set NULL if
invalid
        }
    } else {
        createStmt.setNull(9, java.sql.Types.DECIMAL); // NULL for SALARY if
```

```
not provided
    }

    // Handling JOININGDATE: Convert from long to MySQL DATE format
    if (joiningdate != null) {
        if (joiningdate instanceof Long) {
            // Convert long to java.util.Date
            Date dateObj = new Date((Long) joiningdate);
            // Format the date to 'yyyy-MM-dd'
            SimpleDateFormat dateFormatter = new SimpleDateFormat("yyyy-MM-
dd");
            String formattedDate = dateFormatter.format(dateObj);
            createStmt.setString(8, formattedDate); // Set the formatted date
        } else {
            trace.error("[Create-Groovy] Invalid JOININGDATE format: Expected
Long but got " + joiningdate.getClass());
            createStmt.setNull(8, java.sql.Types.DATE); // Set NULL if the
format is invalid
        }
    } else {
        createStmt.setNull(8, java.sql.Types.DATE); // NULL for JOININGDATE
if not provided
    }

    // STATUS: ACTIVE or INACTIVE
    if (enableValue) {
        createStmt.setString(7, "ACTIVE"); // STATUS
    } else {
        createStmt.setString(7, "INACTIVE"); // STATUS
    }

    // PASSWORD Handling
    if (pass != null) {
        pass.access(new GuardedString.Accessor() {
            public void access(char[] clearChars) {
                createStmt.setString(10, new String(clearChars)); // PASSWORD
            }
        });
    } else {
        createStmt.setString(10, null); // NULL for PASSWORD if not provided
    }

    // The output parameter for USERID (generated by the stored procedure)
    createStmt.registerOutParameter(11, java.sql.Types.VARCHAR);

    // Execute the stored procedure
    createStmt.executeUpdate();

    // Retrieve the generated USERID (either auto-increment or UUID)
    String generatedUserId = createStmt.getString(11); // Get the output
parameter (USERID)
    trace.info("[Create] Created User with USERID::" + generatedUserId);

    // Return the generated USERID as Uid
    return new Uid(generatedUserId);
```

```
} catch (Exception e) {
    trace.error("[Create-Groovy] Error during user creation: " +
e.getMessage());
    throw e;  // Re-throw exception to signal failure
} finally {
    // Clean up resources
    if (createStmt != null) {
        createStmt.close();
    }
}
```

Create Account Stored Procedure

```
DELIMITER $$

CREATE PROCEDURE ADD_PERSON (
    IN USERNAME VARCHAR(50),
    IN FIRSTNAME VARCHAR(50),
    IN LASTNAME VARCHAR(50),
    IN EMAIL VARCHAR(50),
    IN COUNTRYCODE VARCHAR(20),
    IN DESCRIPTION VARCHAR(50),
    IN STATUS VARCHAR(50),
    IN JOININGDATE DATE,
    IN SALARY DECIMAL(10,2),
    IN PASSWORD VARCHAR(50),
    OUT USERID VARCHAR(50)
)
BEGIN
    DECLARE generated_user_id VARCHAR(50);
    DECLARE is_auto_increment VARCHAR(50);  -- Change to VARCHAR to hold
string values like 'auto_increment'

    -- Check if USERID column has the 'auto_increment' flag in the EXTRA
column
    SELECT EXTRA
    INTO is_auto_increment
    FROM INFORMATION_SCHEMA.COLUMNS
    WHERE TABLE_NAME = 'MYDBAT_PERSON'
      AND COLUMN_NAME = 'USERID'
      AND TABLE_SCHEMA = DATABASE();

    -- Check if 'auto_increment' is present in the EXTRA column
    IF is_auto_increment LIKE '%auto_increment%' THEN
        -- If USERID is auto-increment
        INSERT INTO MYDBAT_PERSON (
            USERNAME, FIRSTNAME, LASTNAME, EMAIL, COUNTRYCODE, DESCRIPTION,
STATUS, JOININGDATE, SALARY, PASSWORD
        )
        VALUES (
            USERNAME, FIRSTNAME, LASTNAME, EMAIL, COUNTRYCODE, DESCRIPTION,
STATUS, JOININGDATE, SALARY, PASSWORD
        );
        -- Retrieve the auto-generated USERID and cast it to VARCHAR
        SET USERID = CAST(LAST_INSERT_ID() AS CHAR);
```

```
    ELSE
        -- If USERID is NOT auto-increment, generate a UUID for USERID
        SET generated_user_id = UUID();
        -- Insert the record with the generated UUID
        INSERT INTO MYDBAT_PERSON (
            USERID, USERNAME, FIRSTNAME, LASTNAME, EMAIL, COUNTRYCODE,
DESCRIPTION, STATUS, JOININGDATE, SALARY, PASSWORD
        )
        VALUES (
            generated_user_id, USERNAME, FIRSTNAME, LASTNAME, EMAIL,
COUNTRYCODE, DESCRIPTION, STATUS, JOININGDATE, SALARY, PASSWORD
        );
        -- Set the generated UUID in the output, as a VARCHAR
        SET USERID = generated_user_id;
    END IF;
END$$

DELIMITER ;
```

This stored procedure is a sample implementation for creating a user in the database. The script checks if the `USERID` column of the `MYDBAT_PERSON` table has auto-increment set, which means that the userid is populated automatically. If auto-increment is set then the user is created using a SQL statement without the userid as this is set automatically by the database. If auto-increment is not set then the user is created using a SQL statement which generates the userid and inserts it into the `MYDBAT_PERSON` table. As the implementation of the `USERID` may vary in different implementations, this stored procedure should according to your specific requirements.

Sample Add Child Script

This script is invoked during provisioning of entitlements/permissions to users from Oracle Access Governance. Here we are inserting data into the MYDBAT_PERSON_GROUP and MYDBAT_PERSON_ROLE tables.

Add Child Script

```
import org.identityconnectors.framework.common.objects.*
import java.text.*

trace.info("[addMultiValuedAttributeScript-Groovy] Adding Child data::" +
attributes)
childst = null
try {
    // Adding Group data
    childDataEOSet = null

    // Logic for handling simple multi-valued attributes
    if (attributes.get("MYDBAT_PERSON_GROUP") != null) {
        childDataEOSet = attributes.get("MYDBAT_PERSON_GROUP").getValue()
        childst = conn.prepareStatement("INSERT INTO dbat.MYDBAT_PERSON_GROUP
(USERID, GROUPID) VALUES (?, ?)")
        int id = attributes.get("__UID__").getValue().get(0).toInteger()

        if (childDataEOSet != null) {
            trace.info("[addMultiValuedAttributeScript] Adding Group data.")
```

```
            // Iterate through child data and insert into table
            for (iterator = childDataEOSet.iterator(); iterator.hasNext(); ) {
                eo = iterator.next()
                attrsSet = eo.getAttributes()
                grpattr = AttributeUtil.find("GROUPID", attrsSet)

                if (grpattr != null) {
                    // Extract group ID and insert record
                    groupid = grpattr.getValue().get(0)
                    childst.setInt(1, id)
                    childst.setString(2, groupid)
                    childst.executeUpdate()
                    childst.clearParameters()
                }
            }
        }
    }
} finally {
    if (childst != null)
        childst.close()
}

try {
    childDataEOSet = null
    // Logic for handling complex multi-valued attributes
    if (attributes.get("MYDBAT_PERSON_ROLE") != null) {
        childDataEOSet = attributes.get("MYDBAT_PERSON_ROLE").getValue()
        childst = conn.prepareStatement("INSERT INTO dbat.MYDBAT_PERSON_ROLE
(USERID, ROLEID) VALUES (?, ?)")

        int id = attributes.get("__UID__").getValue().get(0).toInteger()

        if (childDataEOSet != null) {
            trace.info("[addMultiValuedAttributeScript] Adding Role data.")
            for (iterator = childDataEOSet.iterator(); iterator.hasNext(); ) {
                eo = iterator.next()
                attrsSet = eo.getAttributes()
                roleattr = AttributeUtil.find("ROLEID", attrsSet)

                if (roleattr != null) {
                    // Extract role ID and insert record
                    roleid = roleattr.getValue().get(0)
                    childst.setInt(1, id)
                    childst.setString(2, roleid)

                    childst.executeUpdate()
                    childst.clearParameters()
                }
            }
        }
    }
} finally {
    if (childst != null)
        childst.close()
}
```

Sample Remove Child Script

This script is invoked during deprovisioning of entitlements/permissions from users from Oracle Access Governance. Here we are removing data from MYDBAT_PERSON_GROUP and MYDBAT_PERSON_ROLE tables using stored procedures.

Remove Child Script

```
import org.identityconnectors.framework.common.objects.*

trace.info("[removeMultiValuedAttributeScript] Removing Child data::" +
attributes)

try {
    childDataEOSet = null
    delSt = null
    // Get UID and convert to int
    int id = Integer.parseInt(attributes.get("__UID__").getValue().get(0))

    // Handle removal of person group data
    if (attributes.get("MYDBAT_PERSON_GROUP") != null) {
        childDataEOSet = attributes.get("MYDBAT_PERSON_GROUP").getValue()

        // Call the MySQL stored procedure
        delSt = conn.prepareCall("{CALL DELETE_USERGROUP(?, ?)}")

        if (childDataEOSet != null) {
            trace.info("[removeMultiValuedAttributeScript] Removing Group
data.")
            // Iterate through child data and delete
            for (iterator = childDataEOSet.iterator(); iterator.hasNext(); ) {
                eo = iterator.next()
                attrsSet = eo.getAttributes()
                grpattr = AttributeUtil.find("GROUPID", attrsSet)

                if (grpattr != null) {
                    groupid = grpattr.getValue().get(0)
                    delSt.setInt(1, id) // Use setInt for integer ID
                    delSt.setString(2, groupid)
                    delSt.executeUpdate()
                    trace.info("[removeMultiValuedAttributeScript] Deleted
Group::" + groupid)
                }
            }
        }
    }
} finally {
    if (delSt != null)
        delSt.close()
}

try {
    childDataEOSet = null
    delSt = null
    // Get UID and convert to int
```

```
    int id = Integer.parseInt(attributes.get("__UID__").getValue().get(0))

    // Handle removal of person role data
    if (attributes.get("MYDBAT_PERSON_ROLE") != null) {
        childDataEOSet = attributes.get("MYDBAT_PERSON_ROLE").getValue()

        // Call the MySQL stored procedure
        delSt = conn.prepareCall("{CALL DELETE_USERROLE(?, ?)}")

        if (childDataEOSet != null) {
            trace.info("[removeMultiValuedAttributeScript] Removing Role
data.")
            for (iterator = childDataEOSet.iterator(); iterator.hasNext(); ) {
                eo = iterator.next()
                attrsSet = eo.getAttributes()
                roleattr = AttributeUtil.find("ROLEID", attrsSet)

                if (roleattr != null) {
                    rolename = roleattr.getValue().get(0)
                    delSt.setInt(1, id) // Use setInt for integer ID
                    delSt.setString(2, rolename)
                    delSt.executeUpdate()
                    trace.info("[removeMultiValuedAttributeScript] Deleted
Role::" + rolename)
                }
            }
        }
    }
} finally {
    if (delSt != null)
        delSt.close()
}
```

Stored Procedure: Remove Child

Delete User Role

```
DELIMITER $$

CREATE PROCEDURE DELETE_USERROLE (
    IN input_userid INT,
    IN input_roleid VARCHAR(20)
)
BEGIN
    -- Check if the record exists before attempting deletion
    IF EXISTS (
        SELECT 1
        FROM MYDBAT_PERSON_ROLE
        WHERE USERID = input_userid AND ROLEID = input_roleid
    ) THEN
        -- Perform the deletion
        DELETE FROM MYDBAT_PERSON_ROLE
        WHERE USERID = input_userid AND ROLEID = input_roleid;
    ELSE
        -- If no record exists, signal an error or do nothing
        SIGNAL SQLSTATE '45000'
```

```
        SET MESSAGE_TEXT = 'The specified USERID and ROLEID combination does
not exist in MYDBAT_PERSON_ROLE.';
    END IF;
END$$

DELIMITER ;
```

Delete User Group

```
DELIMITER $$

CREATE PROCEDURE DELETE_USERGROUP (
    IN input_userid INT,
    IN input_groupid VARCHAR(20)
)
BEGIN
    -- Check if the record exists before attempting deletion
    IF EXISTS (
        SELECT 1
        FROM MYDBAT_PERSON_GROUP
        WHERE USERID = input_userid AND GROUPID = input_groupid
    ) THEN
        -- Perform the deletion
        DELETE FROM MYDBAT_PERSON_GROUP
        WHERE USERID = input_userid AND GROUPID = input_groupid;
    ELSE
        -- If no record exists, signal an error or do nothing
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'The specified USERID and GROUPID combination does
not exist in MYDBAT_PERSON_GROUP.';
    END IF;
END$$

DELIMITER ;
```

Sample Delete Script

This script is invoked during revocation of an account from Oracle Access Governance. Here we are deleting the data user relationship tables, MYDBAT_PERSON_ROLE and MYDBAT_PERSON_GROUP, as well as data from the MYDBAT_PERSON table

Delete Script

```
import java.sql.PreparedStatement;
import org.identityconnectors.framework.common.objects.*;

// Get the UID from the input map 'attributes'
String uid = attributes.get("__UID__").getValue().get(0);
trace.info("[Delete-Groovy] Deleting user:: " + uid);

try {
    // Delete data from child tables and then, main table
    // Delete user roles
    st = conn.prepareStatement("DELETE FROM dbat.MYDBAT_PERSON_ROLE WHERE
userid=?");
```

```
    st.setString(1, uid);
    st.executeUpdate();
    st.close();

    // Delete user groups
    st = conn.prepareStatement("DELETE FROM dbat.MYDBAT_PERSON_GROUP WHERE
userid=?");
    st.setString(1, uid);
    st.executeUpdate();
    st.close();

    // Delete user account
    st = conn.prepareStatement("DELETE FROM dbat.MYDBAT_PERSON WHERE
userid=?");
    st.setString(1, uid);
    st.executeUpdate();
} finally {
    if (st != null)
        st.close();
};

trace.info("Deleted user:: " + uid);
```

Sample Update Script

This script is invoked during provisioning operations when account is updated from Oracle Access Governance. Here we are updating the data in MYDBAT_PERSON table

Update Script

```
import org.identityconnectors.framework.common.objects.*;
import java.text.*;
import org.identityconnectors.framework.common.exceptions.*;
import java.sql.*;

trace.info("[Update-Groovy] Attributes::" + attributes);

/** During an Update operation, AGCS sends the UID attribute along with
updated attributes. Get all the values of attributes **/
String id = attributes.get("__UID__") != null ?
attributes.get("__UID__").getValue().get(0) : null;
String firstName = attributes.get("FIRSTNAME") != null ?
attributes.get("FIRSTNAME").getValue().get(0) : null;
String lastName = attributes.get("LASTNAME") != null ?
attributes.get("LASTNAME").getValue().get(0) : null;
String email = attributes.get("EMAIL") != null ?
attributes.get("EMAIL").getValue().get(0) : null;
String description = attributes.get("DESCRIPTION") != null ?
attributes.get("DESCRIPTION").getValue().get(0) : null;
String salary = attributes.get("SALARY") != null ?
attributes.get("SALARY").getValue().get(0) : null;
String joindate = attributes.get("JOININGDATE") != null ?
attributes.get("JOININGDATE").getValue().get(0) : null;
Boolean enableValue = attributes.get("__ENABLE__") != null ?
attributes.get("__ENABLE__").getValue().get(0) : true;
```

```
// Throw exception if uid is null
if (id == null) throw new ConnectorException("UID Cannot be Null");

PreparedStatement stmt = null;
try {
    // Create prepared statement to update the MYDBAT_PERSON table
    stmt = conn.prepareStatement("UPDATE dbat.MYDBAT_PERSON SET
FIRSTNAME=IFNULL(?, FIRSTNAME), LASTNAME=IFNULL(?, LASTNAME), EMAIL=IFNULL(?,
EMAIL), SALARY=IFNULL(?, SALARY), JOININGDATE=IFNULL(?, JOININGDATE),
STATUS=IFNULL(?, STATUS) WHERE USERID =?");

    // Set SQL input parameters
    stmt.setString(1, firstName); // First name
    stmt.setString(2, lastName); // Last name
    stmt.setString(3, email);    // Email

    // Handle salary: Convert to BigDecimal if not null, otherwise set SQL
NULL
    if (salary != null) {
        stmt.setBigDecimal(4, new BigDecimal(salary));
    } else {
        stmt.setNull(4, java.sql.Types.DECIMAL); // Set SQL NULL for salary
    }

    // Handle joindate: Convert to MySQL date format if not null
    String dateStr = null;
    if (joindate != null) {
        Date date = new Date(joindate);
        DateFormat targetFormat = new SimpleDateFormat("yyyy-MM-dd"); //
MySQL date format
        dateStr = targetFormat.format(date);
    }
    stmt.setString(5, dateStr); // Joining date

    // Handle enable/disable status
    if (enableValue) {
        stmt.setString(6, "Enabled");
    } else {
        stmt.setString(6, "Disabled");
    }

    // Set UID for the WHERE condition
    stmt.setString(7, id);

    // Execute the update
    stmt.executeUpdate();
} finally {
    // Ensure the statement is closed to release resources
    if (stmt != null) stmt.close();
}

trace.info("[Update] Updated user::" + id);
return new Uid(id);
```

The Oracle Unified Directory connector integrates Oracle Access Governance with Oracle Unified Directory. You can establish a connection between Oracle Unified Directory and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

Preinstall

Before you install and configure an Oracle Unified Directory orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

The system can be any one of the following:

- **Oracle Unified Directory 11g** 11.1.1.5.0, 11.1.2.0.0, 11.1.2.2.0, and 11.1.2.3.0
- **Oracle Unified Directory 12c** 12.2.1.3.0 and 12.2.1.4.0
- **Oracle Unified Directory 14c** 14.1.2.1.0

Supported Operations (Authoritative)

The Oracle Unified Directory orchestrated system supports the following operations where Oracle Unified Directory is the authoritative source for identities:

- Create user
- Update user
- Delete user
- Enable user
- Disable user
- Reset password
- Create group or organization unit
- Update group name or organization unit name
- Delete group or organization unit
- Update container DN
- Add groups
- Revoke groups

Supported Operations (Non-Authoritative)

The Oracle Unified Directory orchestrated system supports the following operations where Oracle Unified Directory is a managed system:

- Create user
- Reset password
- Add groups
- Revoke groups

Create a System User Account for Oracle Unified Directory Orchestrated System Operations

Oracle Access Governance requires a user account to access the system during service operations. Depending on the system you are using, you can create the user in your system and assign specific permissions and roles to the user.

For Oracle Unified Directory:

You must create a system user account for performing the following functions.

- Create, modify, and delete entries related to the managed objects, including accounts, groups, roles (if supported), and organizational units (ou).
- Update passwords for users.

Create an admin user account on the Oracle Unified Directory system. For details of how to do this, see the relevant sections in the following:

- **Oracle Unified Directory 11g**: Configuring Root Users
- **Oracle Unified Directory 12c**: Configuring Root Users

Install

You can establish a connection between Oracle Unified Directory and Oracle Access Governance by entering connection details and configuring your OUD environment. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, specify which type of system you would like to onboard.

1. Select **Oracle Unified Directory** and click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**

- **I want to manage permissions for this system**

The default value in each case is *Selected*.

4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the target Oracle Unified Directory.

1. In the **Host** field, enter the hostname or IP address for the directory you want to integrate with Oracle Access Governance.
2. In the **Port** field, enter the value of the TCP/IP port number used to communicate with the LDAP server.
3. Enter the distinguished name which you will use to authenticate to the directory, in the **Administrator Username** field. This is the user you created in Create a Target System User Account for Oracle Unified Directory Orchestrated System Operations.
4. Enter the password of the target distinguished name in the **Password** field. Confirm the password in the **Confirm password** field.
5. Enter a base context from which to begin searches for users and groups into the **Base Contexts** field.
6. In the **Failover** field, enter a list of failover servers in the format `<servername>:<port>, <servername>:<port>, ...`, for example `OUDExample1:636, OUDExample1:636, ...`
7. In the **SSL Enabled** field, ensure that the value **true** is selected.
8. Check the right hand pane to view **What I've selected.** If you are happy with the details entered, select **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up** where you are prompted to download the agent for your Orchestrated System. Once you have downloaded the agent, you can install and configure the agent in your environment using the instructions in Install Oracle Access Governance Agent.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Postinstall

There are no postinstall steps associated with a Oracle Unified Directory system.

The Oracle Internet Directory connector integrates Oracle Access Governance with Oracle Internet Directory. You can establish a connection between Oracle Internet Directory and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

Preinstall

Before you install and configure an Oracle Internet Directory orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

The system can be any one of the following:

- **Oracle Internet Directory 9.x**
- **Oracle Internet Directory 10.1.4.x**
- **Oracle Internet Directory 11g** 11.1.1.5.0, 11.1.1.6.0, 11.1.1.7.0 and 11.1.1.9.0
- **Oracle Internet Directory 12c** 12.2.1.3.0 and 12.2.1.4.0
- **Oracle Internet Directory 14c** 14.1.2.1.0

Supported Operations

The Oracle Internet Directory orchestrated system supports the following operations:

- Create user
- Reset password
- Add groups
- Revoke groups

Create a System User Account for Oracle Internet Directory Orchestrated System Operations

Oracle Access Governance requires a user account to access the target system during service operations. Depending on the system you are using, you can create the user and assign specific permissions and roles to the user.

For Oracle Internet Directory:

You must create a system user account for performing the following functions.

- Create, modify, and delete entries related to the managed objects, including accounts, groups, roles (if supported), and organizational units (ou).
- Update passwords for users.

Create an admin user, admin group, and ACIs on the OID target system. To perform this task, you must be an administrator on the OID target system who is familiar with command-line

utilities such as `ldapsearch` and `ldapmodify`. If you prefer, you can also use Oracle Directory Services Manager to perform these functions.

For details of how to do this, see the relevant sections in the following:

- **Oracle Internet Directory 11g**: Creating Another Account With Superuser Privileges
- **Oracle Internet Directory 12c**: Creating Another Account With Superuser Privileges

Install

You can establish a connection between Oracle Internet Directory and Oracle Access Governance by entering connection details and configuring your OID environment. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon, select **Service Administration** → **Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, specify which type of system you would like to onboard.

1. Select **Oracle Internet Directory** and click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the target Oracle Internet Directory.

1. In the **Host Name** field, enter the hostname or IP address for the directory you want to integrate with Oracle Access Governance.
2. In the **Port Number** field, enter the value of the TCP/IP port number used to communicate with the LDAP server.
3. Enter the distinguished name which you will use to authenticate to the directory, in the **Administrator Username** field. This is the user you created in Create a Target System User Account for Oracle Internet Directory Orchestrated System Operations.
4. Enter the password of the target distinguished name in the **Administrator Password** field. Confirm the password in the **Confirm password** field.
5. Enter a base context from which to begin searches for users and groups into the **Base Contexts** field.
6. In the **Failover** field, enter a list of failover servers in the format `<servername>:<port>, <servername>:<port>, ...`, for example `OUDExample1:636, OUDExample1:636, ...`
7. In the **SSL Enabled** field, ensure that the value **true** is selected.
8. Check the right hand pane to view **What I've selected.** If you are happy with the details entered, select **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up** where you are prompted to download the agent for your Orchestrated System. Once you have downloaded the agent, you can install and configure the agent in your environment using the instructions in Install Oracle Access Governance Agent.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Postinstall

There are no postinstall steps associated with an Oracle Internet Directory system.

The Oracle e-Business Suite User Management (UM) connector integrates Oracle Access Governance with database user management tables in Oracle Database. You can establish a connection between Oracle e-Business Suite User Management (UM) and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the **Connected Systems** functionality available in the Oracle Access Governance Console.

Preinstall

Before you install and configure a Oracle e-Business User Management (UM) connected system, you should consider the following pre-requisites and tasks.

Certified Components

The target system can be any one of the following:

- **Oracle E-Business Suite 12.1.1 through 12.1.3**
- **Oracle E-Business Suite 12.2.x**

These applications may run on Oracle Database 10g, 11g, 12c, or 19c as either single database or Oracle RAC implementation.

> **Note:**
>
> If your target system is running on Oracle Database release 19.x, then download and apply the Oracle Database patch 31142749 from My Oracle Support. Applying this patch ensures that provisioning operations work fine.

Supported Connector Operations

The Oracle e-Business Employee Reconciliation (HRMS) connected system supports the following connector operations:

- **User Management**
  - Create person
  - Update person
  - Delete person
  - Enable person
  - Disable person
- **Entitlement Grant Management**
  - Add role
  - Update role
  - Remove role
  - Add responsibility
  - Update responsibility
  - Remove responsibility

Create a Target System User Account for Oracle e-Business User Management (UM) Connected System Operations

Oracle Access Governance requires a user account to access the target system that can be used by the connector to perform connector operations. Depending on the target system you are using, you can create the user in your target system and assign specific permissions and roles to the user.

For Oracle e-Business User Management (UM):

1. From the installation media, copy the scripts directory to a temporary directory on either the target system host computer or a computer on which the Oracle Database Client has been installed.
2. On the computer where you copy the scripts directory, verify that there is a TNS entry in the `tnsnames.ora` file for the target system database.
3. Change to the directory containing the scripts directory and depending on the host platform, run either the `Run_UM_DBScripts.sh` or `Run_UM_DBScripts.bat` file. These files are present in the scripts directory of the installation media.
4. When you run the script, you are prompted for the following information:
   - `Enter the ORACLE_HOME`

     Set a value for the `ORACLE_HOME` environment variable. This prompt is displayed only if the `ORACLE_HOME` environment variable has not been set on the computer on which you are running the script.
   - `Enter the System User Name`

     Enter the login (user name) of a DBA account with the privileges to create and configure a new target system user.
   - `Enter the name of the database`

     Enter the connection string or service name given in the `tnsnames.ora` file to connect to the target system database.

     This connects you to the SQL*Plus client.
   - `Enter password`

     Enter the password of the APPS user in the target system. The Type and Package are created, and then the connection to the database is disconnected.
   - `Enter password`

     Enter the password of the dba user.
   - `Enter New database Username to be created`

     Enter a user name for the target system account that you want to create.
   - `Enter the New user password`

     Enter a password for the target system account that you want to create.

     This installs all wrappers packages under the APPS schema, creates the new target system account, and then grants all the required privileges on the tables and packages.
   - `Connecting with newly created database user`

     Enter the connection string or service name that you provided earlier.

The user account for connector operations is created.

Install

You can establish a connection between Oracle Database(DB2) and Oracle Access Governance by entering connection details and configuring your database environment. To achieve this, use the Connected Systems functionality available in the Oracle Access Governance Console.

Navigate to the Connected Systems Page

Navigate to the Connected Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Connected Systems**.
2. Click the **Add a connected system** button to start the workflow.

Add a Connected System

To start the add a connected system workflow, you should select the type of system that you would like to connect with Oracle Access Governance:

Select the **Add** button on the **Connect to an Oracle application**tile.

Select system

On the **Select system** step of the workflow, you can specify which type of application you would like to onboard.

1. Select **Oracle E-Business User Management**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the connected system:

1. Enter a name for the application you want to connect to in the **What do you want to call your Oracle application?** field.
2. Enter a description for the application in the **How do you want to describe this Oracle application?** field.
3. Select the type of source to pull identities into Oracle Access Governance.
   By default, *This is the authoritative source for my Identities* option is selected.
4. Click **Next**.

Configure

On the **Configure** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the target system.

1. In the **User** field, enter the user ID of the DB user account that Oracle Access Governance uses to connect to the Oracle E-Business Suite User Management system. For example, `sys` as `sysdba`.
2. Enter the password of the target database user in the **Password** field. Confirm the password in the **Confirm password** field.
3. In **JDBC URL Template** field, specify the database connection string in the format *host:post:sid* syntax format. For example, `jdbc:oracle:thin:@%host:%port:%sid`. For

more information on JDBC URL formats, refer to the Determining Values for the JDBC URL and Connection Properties Parameters

Finish up

The **Finish Up** step of the workflow allows you to download and install the configured Oracle E-Business User Management agent.

You can establish a connection between Oracle e-Business Employee Reconciliation (HRMS) and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

Preinstall

Before you install and configure a Oracle e-Business Employee Reconciliation (HRMS) orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

The system can be any one of the following:

- **Oracle E-Business Suite 12.1.1 through 12.1.3**
- **Oracle E-Business Suite 12.2.x**

These applications may run on Oracle Database 10g, 11g, 12c, or 19c as either single database or Oracle RAC implementation.

> **Note:**
>
> If your system is running on Oracle Database release 19.x, then download and apply the Oracle Database patch 31142749 from My Oracle Support. Applying this patch ensures that provisioning operations work fine.

Supported Operations

The Oracle e-Business Employee Reconciliation (HRMS) orchestrated system supports the following operations:

- Create person
- Add address
- Update address
- Delete address
- Add assignment
- Update assignment
- Delete assignment

Create a System User Account for Oracle e-Business Employee Reconciliation (HRMS) Orchestrated System Operations

Oracle Access Governance requires a user account to access the system during reconciliation operations. Depending on the system you are using, you can create the user in your system and assign specific permissions and roles to the user.

For Oracle e-Business Employee Reconciliation (HRMS):

1. Download all the files present inside this `https://github.com/oracle/docker-images/tree/main/OracleIdentityGovernance/samples/scripts/Oracle_EBS_HRMS/1.0` location and copy them to a temporary directory on either the system host computer or a computer on which the Oracle Database Client has been installed.
   Alternatively, you can run the following steps to get the scripts:
   a. `wget https://github.com/oracle/docker-images/archive/refs/heads/main.zip`
   b. `unzip main.zip`
   c. `cp docker-images-main/OracleIdentityGovernance/samples/scripts/Oracle_EBS_HRMS/1.0/* TEMP_DIR`
      Where `TEMP_DIR` is a temporary directory on either the system host computer or a computer on which the Oracle Database Client has been installed.
2. On the computer where you copy the scripts directory, verify that there is a TNS entry in the `tnsnames.ora` file for the system database.
3. Change to the directory containing the scripts directory and depending on the host platform, run either the `Run_HRMS_DBScripts.sh` or `Run_HRMS_DBScripts.bat` file. These files are present in this `https://github.com/oracle/docker-images/tree/main/OracleIdentityGovernance/samples/scripts/Oracle_EBS_HRMS/1.0` location.
4. When you run the script, you are prompted for the following information:
   - `Enter the ORACLE_HOME`

     Set a value for the ORACLE_HOME environment variable. This prompt is displayed only if the ORACLE_HOME environment variable has not been set on the computer on which you are running the script.
   - `Enter the System User Name`

     Enter the login (user name) of a DBA account with the privileges to create and configure a new database user.
   - `Enter the name of the database`

     Enter the connection string or service name given in the `tnsnames.ora` file to connect to the system database.
   - `Would you like to create new user for connector operations [y/n]`

     Enter `y` or `n` to specify whether you want to create a new user for connector operations.

     This connects you the SQL*Plus client.
   - `Are you running this script with EBS target 12.1.x [y/n]`

     Enter `y` if you are using Oracle E-Business Suite release 12.1.1 through 12.1.3. When you do so, version compatible scripts will run on your system.

     Enter `n` if you are using Oracle E-Business Suite 12.2.x and later versions.

- `Enter password`

  Enter the password for the Oracle database login. If you entered `n` at the earlier prompt to create a new user for connector operations, then the Type and Package are created, and then the connection to the database is disconnected. If you entered `y`, then the Type and Package are created, and then the connection to the database remains.

- `Enter password`

  Enter the password of the dba user.

- `Enter New database Username to be created`

  Enter a user name for the database account that you want to create.

- `Enter the New user password`

  Enter a password for the database account that you want to create.

  This installs all wrappers packages under the APPS schema, creates the new database account, and then grants all the required privileges on the tables and packages.

- `Connecting with newly created database user`

  Enter the connection string or service name that you provided earlier.

- `Enter the hostname for network acl [Input will be ignored If DB version is earlier than 11g]`

  Enter the name of the computer hosting network acl in the following format:

  `*.`*`DOMAIN_NAME`*`.com`

  This prompt is received only if you entered `y` at one of the earlier prompts to create a new user for connector operations.

Install

You can establish a connection between Oracle e-Business Employee Reconciliation (HRMS) and Oracle Access Governance by entering connection details and configuring your database environment. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration** → **Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of application you would like to onboard.

1. Select **Oracle E-Business Suite HRMS**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to

their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:

- Disable
- Delete

3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the target system.

1. In **JDBC URL Template** field, Specify the jdbc driver connection URL in the format *jdbc:oracle:thin:@[host]:[port]:[DB]*. For more info, refer to the JDBC driver documentation.
2. In the **User** field, enter the database user with permission to access account table.
3. Enter the password of the target database user in the **Password** field. Confirm the password in the **Confirm password** field.
4. In **Delete Person** field, specify whether the employee record must be completely deleted from the target system. When you delete an employee record, the employee record is just set to terminated, but the record is not completely deleted from the target system.
   - If you set the value of this parameter to true, then the employee record is completely deleted from the target system.
   - If you set the value of this parameter to false, then the employee record is not deleted from the target system, but its status is just set to terminated.
5. In **Include Future Hires** field, specify whether the connector must detect and reconcile records with future dated start date values.
   - If you set the value of this parameter to `true`, then the connector reconciles all employee records with future dated Start Date values.

- If you set the value of this parameter to `false`, then the connector does not reconcile employee records with future dated Start Date values.

Default value is `true`.

Finish Up

The final step of the workflow is **Finish Up** where you are prompted to download the agent for your Orchestrated System. Once you have downloaded the agent, you can install and configure the agent in your environment using the instructions in Install Oracle Access Governance Agent.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Postinstall

There are no postinstall steps associated with an Oracle E-Business HRMS system.

Prerequisites

Before you install and configure a Oracle Siebel CRM orchestrated system, you should consider the following prerequisites and tasks.

Certified Components

The system can be any one of the following:

- Siebel CRM 7.5 through Siebel CRM 8.2.2
- Siebel Innovation Pack 2015
- Siebel Innovation Pack 2016
- Siebel Innovation Pack 2017
- Siebel Innovation Pack 2018
- Siebel 19.x, 20.x, 23.x

Supported Modes

Oracle Siebel CRM orchestrated system supports the following modes:

- **Authoritative Source**
- **Managed System**

Supported Operations

The Siebel orchestrated system supports the following operations:

- Create user
- Delete user
- Assign Position

- Revoke Position
- Assign Responsibility
- Revoke Responsibility

Create Siebel User Account

> **Note:**
>
> The system user account for connector operations must be created in the LDAP repository. As a security precaution, ensure that this account does not have access to areas protected by Oracle Access Manager.

To create the user account on Siebel, perform the following:

1. Log in to **Siebel**.
2. Click **Site Map** icon.
3. Click **Administration** → **User**.
4. Click **Employees**.
5. Click **New**.
   Enter the following details for the account that you are creating:
   - Last Name
   - First Name
   - Job Title
   - User ID
   - Responsibility: Select Siebel Administrator.
   - Position: Select Siebel Administrator.
   - Organization: Select Default Organization.
   - Employee Type

   To create the user account on the Siebel database, perform the following:

   a. Open the Siebel home directory.
   b. Open the dbsrvr directory.
   c. Open one of the following directories:
      - For IBM DB2 UDB: DB2
      - For Microsoft SQL Server: MSSQL
      - For Oracle Database: Oracle

   To open one of the following files in a text editor:

   - For IBM DB2 UDB: grantusrdb2.sql
     For Microsoft SQL Server: addusrmsql.sql

     For Oracle Database: grantusroracle.sql

   In the file that you open:

   - Specify the **User ID** of the user that you create in Step 1.

- Set a password for the user.
- Provide other required details.
- Run the script.

Additional Configuration Steps and Guidelines for the Target System

Siebel needs to be configured to use either a database or an LDAP repository to store user information. If an LDAP repository is used, then you must ensure that the following prerequisites are addressed:

If Microsoft Active Directory is used as the LDAP repository, then use the ADSI Security Adapter. Ensure that the Propagate Change attribute of the ADSI Security Adapter is set to False on Siebel.

Manually Making Configuration Changes

Perform the followings tasks to manually make the configuration changes:

1. Log in to Siebel Web Tools.
2. Create Workspace as follows:
   a. Click **Workspace** located next to **Main**.
   b. Click **Create**.
   c. Enter the name for your Workspace and provide comments.
   The workspace is now available under **Main**.
3. Close the window.
4. Open the newly created workspace and locate the **Employee BusComp** option as follows.
   a. Under **Type**, select **Expand Business Component**.
   b. Click**Field**.
   c. From the **Business Component** drop-down, select **Name** and search for the employee.
   d. In the **Fields** option, add a new field with the following attributes:

Attribute	Value
Name	User Status
Join	S_USER
Column	STATUS_CD
Picklist	User Status Picklist
Text Length	30
Type	DTYPE_TEXT

5. Create a child Pick Map for this field as follows:
   a. Expand the option **Field** under **Business component**.
   b. Select **Pick Map**.
   c. Add the following attributes under **Pick Map**.

Attribute	Value
Field	User Status
Picklist Field	Value

6. Navigate to Employee List Applet option as follows.
   a. Expand Applet, and select **List**.
   b. Under the **Applet** drop-down list, select **Name** and search for Employee List.
7. In the **List Column** under List, you must add a new list column with the following attributes:

Attribute	Value
Name	User Status
Field	User Status
Available	TRUE
Display Name – String Reference	SBL_USER_STATUS-1004233658-7EI
Display Name	User Status
HTML Display Mode	EncodeData
HTML List Edit	TRUE
HTML Row Sensitive	TRUE
HTML Type	Field
Runtime	TRUE
Text Alignment	Left
Show in List	TRUE
Text Alignment-Label	Left

8. For the same applet, choose the **Edit List Applet Web Template** to add the newly created list column to any empty placeholder in the list as follows:
   a. Expand Applet and select **Applet Web Template**.
   b. Under **Applet Web Template**, choose an empty place holder in **Edit List** and select Edit.
   c. Click **Controls/Columns**, and deselect the option *show unmapped controls only* and select **User Status**
9. Unit test the changes:
   a. Open the Siebel Call Center to **Open** and **Inspect** the workspace for ensuring that the newly added column **User Status** appears in the user interface to change from **Active** to **Inactive**and oppositely.
   b. Change the status to **Inactive** for different known users.
   c. Log out.
   d. Try to log in as other user.

> **Note:**
>
> This test should fail.

10. Deliver the workspace.
    a. Log in to Siebel Web Tools.
    b. Click the Workspace dashboard button and click **Open**.
    c. Click **Version** to provide the comments and create the version.
    d. Click **Submit** and submit the delivery.

e. Click **Deliver** to provide the comments and deliver the workspace.

Importing SIF File

Importing SIF File option allows a developer to make changes (without the manual modifications described above) by importing an archive file (SIF) containing the repository changes. Importing SIF file helps in enabling the status features.

> **Note:**
>
> - By default, this option is disabled.
> - The steps to import a SIF file must be followed if you are not performing the manual configuration steps provided in **Manually Making Configuration Changes** section.

Perform the following steps:

1. In Siebel Web Tools, create a **Developer Workspace** under a upcoming release branch (Integration Workspace).
   a. Click Workspace dashboard option located next to **Main**.
   b. Click **Create.**
   c. Enter the name of your Workspace and provide comments to create the workspace. The workspace is now visible under Main.
   d. Open the newly created workspace.
2. Select **Archive** > **Import from Archive** menu item.
3. Follow the wizard to import the file.
4. Checkpoint and submit the workspace for delivery, rebasing if necessary.
5. Deliver the workspace as follows:
   a. Click the Workspace dashboard option and select your workspace then click **Open**.
   b. Click **Version** to provide your comments and create the version.
   c. Click **Submit** and submit for delivery.
   d. Click **Deliver** to provide the comments and deliver the workspace.
6. Test the changes by following the below:
   a. Open the Siebel Call Center and click **Open** to inspect the workspace for ensuring that the newly added column is visible under **User Status** in the user interface and can be changed from **Active** to **Inactive** and the opposite.
   b. Change the status to **Inactive** for different known users.
   c. Log out.
   d. Try to log in as other user.

   > **Note:**
   >
   > This test should fail.

Configure

You can establish a connection between Oracle Siebel CRM and Oracle Access Governance by entering connection details. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard.

1. Select **Siebel**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the Siebel system.

1. In the **GatewayServer** field, enter the name of the gateway server. A gateway server is a Windows service or UNIX daemon process that stores component definitions and assignments, operational parameters, and connectivity information.
   Sample value:

   ```
   phoenix200458.myapp.myfusionapp.example.com
   ```

2. In the **ServerPort** field, enter the port number at which the target system is listening.
3. In the **User name** field, enter the User ID of the target system user account that you want to use for connector operations.
   Sample value: johnsmith
4. In the **Password** field, enter the password of the target system user account that you want to use for connector operations and confirm the password.
5. In the **Object manager** field, enter the name of the object manager. The term *Object Manager* refers to any of several Siebel Server components that support users accessing Siebel Business Applications through the Siebel Web Client and a Web server.
   A different Siebel Application Object Manager component is provided for each base application among the Siebel Business Applications or Siebel Industry Applications.

   > **Note:**
   >
   > Separate Siebel Application Object Managers are provided for each installed language in which you can run your Siebel applications.

   For example, you can refer to any one of the following for the specific language:

   For English:

   ```
   SCCObjMgr_enu
   ```

   For Brazilian Portuguese:

   ```
   SCCObjMgr_ptb
   ```

   For French:

   ```
   SCCObjMgr_fra
   ```

   For German:

   ```
   SCCObjMgr_deu
   ```

For Italian:

```
SCCObjMgr_ita
```

For Japanese:

```
SCCObjMgr_jpn
```

For Korean:

```
SCCObjMgr_kor
```

For Simplified Chinese:

```
SCCObjMgr_chs
```

For Spanish:

```
SCCObjMgr_esp
```

For Traditional Chinese:

```
SCCObjMgr_cht
```

.

6. In the **Siebel server** field, enter the name of the system server
Sample value:

```
SBA_SIEBEL
```

7. In the **Version** field, enter the version of the system supported by this connector.
Sample value: 15.5

> **Note:**
>
> If the system version that you are using is Siebel CRM 7.5.x or 7.5.x.x then enter 7.5 only as the value of this parameter. For example, if you are using Siebel CRM 7.5.3.7 as the target system, then enter 7.5.

8. In the **Trusted token** field, enter the trusted token value that you specify while configuring the system to communicate with the SSO system. If you have not configured SSO authentication, then enter No.
Sample value: No
9. In the **Enterprise server** field, enter the name of an enterprise, which is a logical collection of Siebel servers that access a single database server and file system.
Sample value: siebel
10. In the **User Type** field, you can specify one of the following Siebel user types:
    - Employee: This user is an internal employee and this user is associated with a position in a division within your company.

- User: This user is also a self-registered partner having no position in your company. However, this user has a responsibility that specifies the application views the user can access.

11. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up**, where you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Postinstall

There are no postinstall steps associated with an Oracle Siebel CRM system.

Prerequisites

Before you install and configure an Oracle NetSuite orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

The system can be any one of the following:

Table 5-22 Certified Components

Component Type	Component
Target System	Oracle NetSuite Release 2023.1
Target API Version	NetSuite v1 and NetSuitePort_2022_1

Supported Modes

Oracle NetSuite orchestrated system supports the following modes:

- **Managed System**

Supported Operations

The Oracle NetSuite orchestrated system supports the following operations:

- Create user
- Delete user
- Reset Password
- Assign Roles to a user
- Revoke Roles from a user
- Assign Group to a user
- Remove Group from a user

Configuring NetSuite System to Perform Operations

This is a high-level summary of the tasks to be performed on the target system before you create the application.

Pre-installation for the NetSuite connector involves performing a series of tasks on the NetSuite system.

Pre-installation involves the following tasks:

1. Login to Oracle NetSuite.
2. Go to ***Setup > Company > Enable Features*** .
3. Click **SuiteCloud** sub-tab and enable the following features from the respective menu items.
    a. **SuiteBuilder**
    Enable the following boxes:
        i. ITEM OPTIONS
        ii. CUSTOM RECORDS
        iii. ADVANCED PDF/HTML TEMPLATES
        iv. REMOVE PERSONAL INFORMATION
    b. **SuiteScript**:
        i. CLIENT SUITESCRIPT
        ii. SERVER SUITESCRIPT
    c. **SuiteFlow**
        i. SUITEFLOW
    d. **SuiteGL**
        i. CUSTOM GL LINES
        ii. CUSTOM TRANSACTIONS
        iii. CUSTOM SEGMENTS
    e. **SuiteBundler**
        i. CREATE BUNDLES WITH SUITEBUNDLER
    f. **SuiteTalk**
        i. SOAP WEB SERVICES
        ii. REST WEB SERVICES
    g. **Manage Authentication**
        i. SUITESIGNON
        ii. TOKEN-BASED AUTHENTICATION
        iii. OAUTH 2.0
    h. **SuiteCloud Development Framework**
        i. SUITECLOUD DEVELOPMENT FRAMEWORK
4. Click **SAVE**.

To create an integration record for an application, follow the below steps:

- 1. Go to **Setup > Integration> Manage Integration > New**.
  2. Enter a name for your application in the Name field.
  3. Enter a description in the Description field, if preferred.
  4. Select Enabled in the State field.
  5. Enter a note in the Note field, if preferred.
  6. On the **Authentication** tab, check the appropriate boxes for your application:
     a. Token-based Authentication
        – i. TOKEN-BASED AUTHENTICATION
          ii. TBA: AUTHORIZATION FLOW
          iii. Define the CALLBACK URL.
     b. O-Auth 2.0
        i. AUTHORIZATION CODE GRANT
        ii. Scope
          i. RESTLETS
          ii. REST WEB SERVICES
        iii. Provide a valid REDIRECT URI
  7. Click **SAVE**.
  8. Ensure to copy the Client Credentials details that will appear on the screen as it is one-time display.
     For Example:

     consumerKey = "fcb9ec7e7d386fab36566e9c4159bXXXXXXX2875841d828aee7e"

     consumerSecret = "bd7780d4396715f5f4586d874379XXXXXX38c42a525c95f70"

To create and assign a Token Based Authentication token:

1. Log in as a user with the **Access Token Management** permission.
2. Go to **Setup > Users/Roles > Access Tokens.**
3. On the Access Tokens page, click **New Access Token**.
4. On the Access Token page:
   a. Select the **Application Name**.
   b. Select the **User**.
   c. Select the **Role**.
   d. The **Token Name** is already populated by default with a concatenation of Application Name, User, and Role. Enter your own name for this token, if preferred.
5. Click **Save**.
6. Ensure to copy the Token details that will appear on the screen as it is one-time display.
   For example:
   tokenId = "0948d37f7XXXXXXXXXXXXXX8075";

   tokenSecret = "86b7bb19cXXXXXXXXabfa0eb401e2c2c24b”

OAuth2.0 Flow to Generate the User-Level Tokens

To generate the user-level access and refresh tokens, there are two steps you must complete manually, and these values should be provided in authToken in Oracle NetSuite Connector basic configuration for authentication.

The following steps must be completed by users who are opting in for Authorization Code Grant:

You must pass the Authorization code grant URL in the internet browser or use Postman to generate the tokens.

1. **Requesting the Authorization Code**

> **Note:**
>
> The token URI for the developer environment is as follows:
> `https://<host name>/services/rest/auth/oauth2/v1/token.`

   a. Enter the following URL in a browser as provided in the example.
   Example:

   ```
   https://<host name> /app/login/oauth2/authorize.nl?
   redirect_uri={callback}&response_type=code&scope=restlets+rest_webservices
   &state=ykv2XLx1BpT5Q0F3MRPHb94j&client_id={ConsumerKey}.
   ```

   Replace `{ConsumerKey}` with your Consumer key / Client id and `{callback}` with your redirect URI. The URL above includes the signature scope required for the eSignature REST API.

   This URL opens the Oracle NetSuite authentication screen.

   b. After you enter your Oracle NetSuite account email address and password and give consent for the requested scopes and then once you redirect to the login Browser Enter the user Credentials to Login and authenticate then Click on the Continue to allow Oracle NetSuite to access your information to Provide the code. The browser will redirect to your redirect URI with a long string returned for the code parameter embedded in the URL.
   Request:

   ```
   https://<host name>/app/login/oauth2/authorize.nl?redirect_uri=http://
   example.com&response_type=code&scope=restlets+rest_webservices&state=ykv2X
   Lx1BpT5Q0F3MRPHb94j&client_id=7e1c238e-xxxx-xxxx-xxxx-abcea08a3171
   ```

   Response: `https://example.com/?
   state=ykv2XLx1BpT5Q0F3MRPHb94j&role=3&entity=4622&company=TSTDRVXXXXXX&cod
   e=096835b6aced….....457b00e3c`

2. **Generating Refresh Tokens Using the Code Generated in Step 1**

   a. To request a refresh token, send a POST request containing your authorization code to the NetSuite authentication service.

   b. Paste the values of Consumer Key and Consumer secret key as User name and Password respectively under Authorization in the Refresh token request with the type as Basic Auth in Postman.

   c. In addition, the refresh token request contains a set of body parameters namely grant_type and code.

i. Update the key as code with value <code>.

> **Note:**
>
> <code> is nothing but the authorization code that you received from the callback in step 1.
> For example, code=096835b6aced.....457b00e3c.

ii. Similarly, update one or more body parameter with the key as `grant_type` and value as `authorization_code` and another body parameter with key as `redirect_uri` and value as the same provided in the step 1.

d. Execute the Authorize Code Grant Refresh Token request to generate an access token and a refresh token.

i. In the response, you will get elements, namely, access_token, token_type, refresh_token, and expires_in.

ii. Copy/save the values of refresh_token.

For more information about how to get a refresh token with Auth Code Grant, see NetSuite Applications Suite.

Examples:

Request:

```
curl --location --request POST " https://<host name>/services/rest/auth/
oauth2/v1/token"--header "Authorization: Basic
N2UxYzIzOGU1Zj........GI3Njg3MzMzMTZm" --header "Content-Type:
application/x-www-form-urlencoded" --data-urlencode
"code=34e8dec4289........a52fe26" --data-urlencode "redirect_uri=https://
example.com" --data-urlencode "grant_type=authorization_code"
```

Response:

```
{ "access_token":"eyJ0eXAi......mX9f7k1g", "token_type":"Bearer",
"refresh_token":"eyJ0eXAi......mruC5c3A", "expires_in":3600 }
```

Table 5-23 Required element for OAuth2.0 authentication

Element	Description
refresh_token	A token that is used to obtain a new access token without requiring user consent and Use this token in the Authorization header of all NetSuite API calls. Providing Values for NetSuite Connector Basic Configuration. After you have obtained the refresh_token value, you must provide these values in authToken under NetSuite Connector basic configuration. For information about configuration, see Configuring the NetSuite Connector. For example, eyJ0eXAi.....mX9f7k1g
refresh_token value	The full refresh token value that is received from authentication.

Configure

You can establish a connection between Oracle NetSuite and Oracle Access Governance by entering connection details. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard.

1. Select **NetSuite**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the Oracle NetSuite instance.

1. In the **Host** field, enter the host name of the system on which your NetSuite application is running.
   Sample Value : `TSTDRVXXXXXXXX.suitetalk.api.netsuite.com`
2. In the **Account name** field, enter the name for the account created on the NetSuite application to perform operations.
   Sample Value : `TSTDRVXXXXXXXX`
3. In the **Consumer key** field, enter the consumerKey.
   Sample Value : `7e1c238e538bafXXXXXXXXbcea08a3171`
4. In the **Consumer secret** field, enter the consumerSecret.
   Sample Value : `fff0b23810704056XXXXXXXXXX0b768733316f`
5. In the **Token ID** field, enter the token Id.
   Sample Value : `3e23ecc14bc7dXXXXXXXd400e56177ed`
6. In the **Token secret** field, enter the Token secret.
   Sample Value : `cd750404ee67653aXXXXXXXXXX646422da64c`
7. In the **Auth URL** field, Enter the URL of the authentication server that validates the client ID and client secret for your system.
   Default value : `/services/rest/auth/oauth2/v1/token`
8. In the **Auth token** field, enter the Refresh Token Values. This value can be fetched by performing OAuth code authorization flow.
   Sample value :
   `eyJ0eXAiOiJNVCIsImFsZyI6IlJTMjU2Iiwia2lkIjoiNjgxODVmZjEtNGU1MS00Y2U5LWFmMWMtNjg5ODEyMjAzMzE3In0.AQoAAAABAAUABwCA8Kx7sbjaSAgAgDDQifS42kgCAGcjU3expKxCtXXXXXXXXXXXXFAAAADQAkAAAANDdhZWE4OWQtNWViYy00NmMyLWI0YmYtNjE5MDRhMjE0MTE1IgAkAAAANDdhZWE4OWQtNWViYy00NmMyLWI0YmYtNjE5MDRhMjE0MTE1MACABwhGsbjaSDcAC1hTwTsYB0GKF0Qif6kfLg.Lk45d4mcBPIrBghYun1S2pVa0EE0XHYTU66cqWpEuPMgSieVTRgwF3wyTOSgyPuiJNf18QTJcG6js4LvVL7sPw8IJwQ6bd`
9. In the Port field, enter the port number the target system is listening on.
   Sample value: `443`
10. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up**, where you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with an Oracle NetSuite system.

Prerequisites

Before you install and configure an Eloqua orchestrated system, you should consider the following prerequisites and tasks.

Certified Components

The system can be any one of the following:

- **Eloqua**

Supported Modes

Eloqua orchestrated system supports the following mode(s):

- **Managed System**

Supported System Operations

The Eloqua orchestrated system supports the following operations:

- Create user
- Delete user
- Reset Password
- Assign Groups to a user
- Remove Groups from a user
- Assign Licences to a user
- Remove Licences from a user

Configure

You can establish a connection between Eloqua and Oracle Access Governance by entering connection details. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard.

1. Select **Eloqua**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to

their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:

- Disable
- Delete

3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the Eloqua application instance.

1. In the **User name** field, enter the username of the system user that you created for performing operations.
   Username should be in the format `<Company name>\<User name>`.
2. In the **Host** field, enter the host name of the machine hosting your Eloqua system.
3. In the **Port** field, enter the port number at which the Eloqua system is listening.
4. Enter the password of the user of the Eloqua system that you created for performing operations, into the **Password/Confirm password** fields.
5. If the Eloqua system requires SSL connectivity, then set the value of this parameter to true in the **SSL Enabled** field. Otherwise set the value to false.
6. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up**, where you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**

- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with an Eloqua system.

Prerequisites

Before you install and configure an Oracle Primavera orchestrated system, you should consider the following prerequisites and tasks.

Certified Components

The system must be the following:

- Oracle Primavera P6 Enterprise Project Portfolio Management / Unifier Cloud Service - Version 20.1 and later

Supported Modes

Oracle Primavera orchestrated system supports the following mode:

- **Managed System**

Supported Operations

The Oracle Primavera orchestrated system supports the following operations. These operations may be initiated during provisioning of the Oracle Primavera system from Oracle Access Governance.

- Create User
- Reset Password
- Assign Roles
- Revoke Roles

Default Supported Attributes

The Oracle Primavera orchestrated system supports the following default attributes. These attributes are mapped depending on the direction of the connection, for example:

- Data being ingested by Oracle Access Governance from Oracle Primavera: `Identity.firstName` will map to `Account.firstName`
- Data being provisioned into Oracle Primavera from Oracle Access Governance: `Account.lastName` will map to `Identity.lastName`

Table 5-24 Default Attributes - Managed System Mode

Primavera Entity	Attribute Name On Managed System	Oracle Access Governance Account Attribute Name	Oracle Access Governance Account Attribute Display Name
User	id	uid	Unique Id
	loginId	name	User Login
	password	password	Password

Table 5-24 (Cont.) Default Attributes - Managed System Mode

Primavera Entity	Attribute Name On Managed System	Oracle Access Governance Account Attribute Name	Oracle Access Governance Account Attribute Display Name
	firstName	firstName	First Name
	lastName	lastName	Last Name
	userType	userType	Person Type
	status	status	Status
	emailAddress	email	Email
	Company	company	Company

Default Matching Rule

The default matching rule for Oracle Primavera orchestrated system

is:

Table 5-25 Default Matching Rules

Mode	Default Matching Rule

Managed System	**Screen value**:
	`User login = Employee user name`
	Attribute name:
	`Account.name = Identity.userName`

Create Oracle Primavera User Account

To create a user account for the Oracle Primavera Orchestrated System perform the following steps:

1. Create a user account on the Orchestrated System with the following:
   - User name
   - Password
   - Host
   - Port
   - TenantId
2. Assign necessary privileges such as, `Cloud administrator` and `P6` to the same user to perform the integration operations.

Configure

You can establish a connection between Primavera and Oracle Access Governance by entering connection details. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon, select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field.

1. Select **Primavera**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the required Oracle Primavera instance.

> **Note:**
>
> The following fields are mandatory, except for the Port field.

1. In the **User Name** field, enter the username of a user you have created on the Primavera system for performing integration operations.
   Sample Value : `johnsmith`
2. In the **Password** field enter the password of the user on your Primavera system.
   Sample Value : `password`
3. In the **Confirm password** field, confirm the password of the user on your Primavera system.
4. In the **Host** field, enter the host name of the machine hosting your system.
   Sample value: `myhost.example.com`
5. In the **Port** field, enter the port number the system is listening on.
6. In the **TenantId** field, enter the value for TenantID. This information is mandatory and the information specified is used for calling the API details.
7. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up**, where you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with an Oracle Primavera system.

Oracle Fusion Cloud Applications

Overview: Integrate Oracle Access Governance with Oracle Fusion Cloud Applications

Oracle Access Governance can be integrated with Oracle Fusion Cloud Applications enabling identity orchestration, including on-boarding of identity (user) data, worker information, and provisioning of Oracle Human Capital (HCM) and Oracle Enterprise Resource Planning (ERP) accounts.

Oracle Fusion Cloud Applications provides enterprise Human Capital Management (HCM) and Enterprise Resource Planning (ERP) functionality. Oracle Access Governance supports the following elements within Oracle Fusion Cloud Applications:

- Oracle Fusion Cloud Applications HCM and Oracle Fusion Cloud Applications ERP as an authoritative (trusted) source of identity information allowing for reconciliation of employees created or modified in Oracle Fusion Cloud Applications.
- Oracle Fusion Cloud Applications as a Managed System enabling provisioning of HCM and ERP application accounts.

Oracle Fusion Cloud Applications Integration Architecture Overview

The integration of Oracle Fusion Cloud Applications allows for retrieving identity data and transferring the data to Oracle Access Governance. Once a connection is established, you can perform provisioning and remediation tasks which are visible in the Managed System.

Oracle Fusion Cloud Applications works with the Fusion Apps API to gain access to Oracle Fusion Cloud Applications through the REST API endpoints. This allows Oracle Fusion Cloud Applications to perform create, read, update, and delete operations on Oracle Access Governance.

- If you select the Authoritative Source mode, you can set up a Oracle Fusion Cloud Applications Orchestrated System, which then allows Oracle Access Governance to retrieve identity data from Oracle Fusion Cloud Applications as an authoritative (trusted) source of identity information.
- If you select the Managed Systems configuration mode, then Oracle Access Governance will allow you to manage HCM and ERP user profile records in Oracle Fusion Cloud Applications. This enables the provisioning of new accounts in Oracle Fusion Cloud Applications from Oracle Access Governance.

Oracle Fusion Cloud Applications Integration Functional Overview

Oracle Fusion Cloud Applications integration supports both Oracle Human Capital (HCM) and Oracle Enterprise Resource Planning (ERP) modules including configuration of the Orchestrated System, user account creation, revocation, change password, and assigning and removal of roles.

Configure Oracle Fusion Cloud Applications Orchestrated System

The first task you need to carry out is to set up and configure Oracle Fusion Cloud Applications Orchestrated System. This gives Oracle Access Governance the details for how to connect to the Oracle Fusion Cloud Applications system from which you want to load data, or manage permissions. Optionally you can configure further elements of the Orchestrated System before running the initial dataload including:

- Notification Settings
- Identity/Account Matching Rules
- Apply data transformations to inbound and outbound data
- Identity attributes

Load Data

After setting up and verifying your Orchestrated System, you can ingest identity and account details from Oracle Fusion Cloud Applications, depending on the configuration mode you have selected, *Authoritative Source* or *Managed System*.

Authoritative Source mode consists of user data from the Oracle Fusion Cloud Applications HCM and ERP modules. If the user is new, then a new identity is created in Oracle Access Governance. If the identity already exists in Oracle Access Governance, then any updates initiated in the Oracle Fusion Cloud Applications system is applied.

Managed System mode comprises of user account data and worker information roles from Oracle Fusion Cloud Applications for HCM and ERP. If the account is new, then a new account is created in Oracle Access Governance together with the associated roles, These roles will be created in Oracle Access Governance as permissions. Accounts and permissions loaded from Oracle Fusion Cloud Applications can be managed by Oracle Access Governance. You can

remediate permissions associated with a managed system account. If the account only has one permission assigned then remediation of this permission will also result in the revoking of the account.

Create Account

As an Oracle Access Governance user you can request access to resources and roles provided in Request Access .

The following ways allows you to create an user account in Oracle Access Governance:

- Ingestion of user records as data from Oracle Fusion Cloud Applications.
- When a role, policy, or access bundle containing Oracle Fusion Cloud Applications roles is assigned to an identity. If you have an identity in Oracle Access Governance then you can request an account by using the **Request a new access** functionality in the Oracle Access Governance console. If you make an access request for an access bundle, or role, after approval, a provisioning operation is initiated. The provisioning process will, if there is not an existing account managed by Oracle Access Governance, create an account on the Oracle Fusion Cloud Applications instance. If an account managed by Oracle Access Governance already exists, then the Oracle Fusion Cloud Applications roles for that account are updated based on the values in the access bundle.

Change Password

The ability to change an account password is provided by the **My Access** functionality in Oracle Access Governance Console. If you change the account password in this page, the details will be sent to the Oracle Fusion Cloud Applications instance in the next provisioning operation.

For more details, refer to Change Account Password.

Assign Permissions using Security Context

Oracle Access Governance users can request access to resources and roles provided in Request Access . You can assign permissions to a Oracle Fusion Cloud Applications account using the **Request a new access** functionality of Oracle Access Governance. This allows you to request an access bundle containing permissions with security details to roles on the Oracle Fusion Cloud Applications system. For details on managing role and policies, see Manage Roles and Manage Policies.

Oracle Access Governance supports the following Security Contexts when integrated with Oracle Fusion Cloud Applications ERP:

- Business Units
- Asset Book Value
- Ledgers or Ledger Sets
- Reference Data Sets
- Data Access Sets
- Inventory Organization
- Intercompany Organization
- Cost Organization
- Manufacturing Plant

When you request an access bundle in Oracle Access Governance for a role, a provisioning operation is initiated which updates the roles in your Oracle Fusion Cloud Applications for the following types of scenarios:

Creating Permission using Security Context during Policy Creation

While creating a policy with Oracle Access Governance for the following use cases:

- Create a new access bundle that has permission with security context and which is already associated with identity collection for the policy.
- Create a new access bundle that has permission with security context and which is already associated with identity collection for the policy. This is applicable in situations when the user already has the access bundle assigned with same permission, but with a different security context.

Editing Permission for Removal of Security Context

You can edit the permissions entitlement using Oracle Access Governance for the following cases:

- Edit the access bundle that have permission with security context to change the security context from permission entitlement which is already associated with an identity collection for the associated policy.
- Edit the access bundle that have permission with security context to remove security context from permission entitlement which is already associated with identity collection via policy.

Remove Permissions

You can remove permissions with the security context from an account by revoking the permissions from the role, policy or access bundle to which it is assigned.

Prerequisites

Before you install and configure an Oracle Fusion Cloud Applications orchestrated system, you should consider the following pre-requisites and tasks.

You must certify your Oracle Fusion Cloud Applications system to access Oracle Access Governance. Refer to Certified Components for details of the versions supported.

Configure

You can establish a connection between Oracle Fusion Cloud Applications and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select and configure a new Orchestrated System

To start the add a orchestrated system workflow, you should select the type of system that you would like to connect with Oracle Access Governance:

Select system

On the **Select system** step of the workflow, you can specify which type of application you would like to onboard.

1. Select **Oracle Fusion Cloud Applications**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

If you are managing permissions with this orchestrated system then an additional option is displayed to enable Segregation of Duties check based on the risk management and compliance feature of Oracle Fusion Cloud Applications. To enable this option for your orchestrated system select **Enable Risk Management and Compliance (RMC) integration for separation of duties check**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Oracle Fusion Cloud Applications Service Account Settings

Grant the following permissions for the below given HCM and ERP service account:

Table 5-26 Role/Permissions for HCM Service Account

Role Name	Role Code
IT Security Manager	ORA_FND_IT_SECURITY_MANAGER_JOB
Integration Specialist	ORA_FND_INTEGRATION_SPECIALIST_JOB
Application Implementation Consultant	ORA_ASM_APPLICATION_IMPLEMENTATION_CONSULTANT_JOB

Table 5-27 Role/ Permissions for ERP Service Account

Account Type	Value
IT Security Manager	ORA_FND_IT_SECURITY_MANAGER_JOB
Integration Specialist	ORA_FND_INTEGRATION_SPECIALIST_JOB
Application Implementation Consultant	ORA_ASM_APPLICATION_IMPLEMENTATION_CONSULTANT_JOB
Access Request Security Administrator	ORA_GTG_ACCESS_REQUEST_SECURITY_ADMINISTRATOR_JOB

The following permission must be granted for **Access Request Security Administrator** role type:

1. Log in to Oracle Fusion Application.
2. Go to **My Enterprise > Setup and Maintenance**.
3. Click **Tasks**.
4. Click **Search** and select **Manage Standard Lookups**.
5. Add the new lookup Type **FUN_DS_OPTIN_OPTIONS** by using the following lookup Code **FUN_DS_GET_BOOKCODE**.

Integration Settings

On the next step of **Integration Settings** of the workflow, enter the details of the applications required to allow Oracle Access Governance to connect to the target Oracle Fusion Cloud Applications.

> **Note:**
>
> For the Oracle Fusion Cloud Applications accounts that are created prior, the default value for **Connected System** would be set to Oracle Human Capital Management (HCM). To change the value to Oracle Enterprise Resource Planning (ERP), or to set to both, follow the changes in the **Integration Settings** below.

1. Determine which application type to integrate with Oracle Access Governance by selecting any of the following application types options.
   - **Both**
   - **Oracle Human Capital Management (HCM)**
   - **Oracle Enterprise Resource Planning (ERP)**

2. In the **Host** field, enter the host name of the machine hosting your source Oracle Fusion Cloud Applications system. Your URL is structured in `https://<host name>:<port>/fcsUI/faces/FuseWelcome`, enter host name.
3. In the **Port** field, enter the port number at which the source Oracle Fusion Cloud Applications system is listening. Your URL is structured in `https://<host name>:<port>/fcsUI/faces/FuseWelcome`, enter port.
4. In the **Username** field, enter the username of the user created on the source for performing orchestrated system operations.
5. In the **Password/Confirm Password** fields, enter the password of the user created on the source for performing orchestrated system operations.
6. Click **Add** to create the orchestrated system.

Post Configuration

There are no postinstall steps associated with a Oracle Fusion Cloud Applications system.

Oracle Fusion Cloud Applications Components Certified for Integration with Oracle Access Governance

The Oracle Fusion Cloud Applications to integrate with Orchestrated System is as follows.

Certified Components

Table 5-28 Certified Components

Component Type	Component
System	The versions of Oracle Fusion Cloud Applications you can use for Oracle Access Governance are: • Oracle Fusion Cloud Applications 24C (11.13.24.07.0) or later

Supported Configuration Modes for Oracle Fusion Cloud Applications

You can use Oracle Access Governance integrations to set up different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Supported Modes

The Oracle Fusion Cloud Applications Orchestrated System supports the following modes:

- **Authoritative Source**
  You can use Oracle Fusion Cloud Applications as an authoritative (trusted) source of identity information for Oracle Access Governance.
- **Managed System**
  You can manage Oracle Fusion Cloud Applications user profile records in Oracle Fusion Cloud Applications including Role and Permission List assignments to these records.

Supported Operations When Provisioning To Oracle Fusion Cloud Applications

To provision an account from Oracle Access Governance to Oracle Fusion Cloud Applications there are certain operations that are supported.

The Oracle Fusion Cloud Applications Orchestrated System supports the following account operations when provisioning a user:

- Create User
- Update User
- Change Password
- Add Roles
- Remove Roles

Default Supported Attributes

Oracle Fusion Cloud Applications Supported Attributes

Oracle Access Governance supports the following default Oracle Fusion Cloud Applications attributes.

- Data with minimum attribute set being ingested by Oracle Access Governance from Oracle Fusion Cloud Applications HCM with support from JML.
- Data with minimum attribute set of person record being ingested by Oracle Access Governance from Oracle Fusion Cloud Applications HCM and ERP modules.

Table 5-29 Default Attributes - Authoritative Source for HCM with JML Support

Attribute Name on Oracle Fusion Cloud Applications	Attribute Name on Oracle Access Governance	Display Name on Oracle Access Governance	Applicable to HCM/ ERP/ BOTH
userName	name	Name	Both
displayName	displayName	Display Name	Both
name.familyName	lastName	Last name	Both
name.givenName	firstName	First name	Both
emails.value	email	Email	Both
emails.type	emailType	Email Type	Both
active	status	Status	Both
workerInformation.personNumber	personNumber	Person Number	Both
workerInformation.manager	manager	Manager	Both
workerInformation.job	jobCode	Job Code	Both
workerInformation.department	department	Department	Both
workRelationships~assignments~BusinessUnitId	businessUnit	Business Unit	HCM
preferredLanguage	preferredLanguage	Preferred Language	HCM
legislativeInfo~Gender	gender	Gender	HCM
PersonId	personId	Person Identification	HCM

Table 5-29 (Cont.) Default Attributes - Authoritative Source for HCM with JML Support

Attribute Name on Oracle Fusion Cloud Applications	Attribute Name on Oracle Access Governance	Display Name on Oracle Access Governance	Applicable to HCM/ ERP/ BOTH
names~EffectiveStartDate	startDate	Start Date	HCM
workRelationships~WorkerType	workerType	Worker Type	HCM
workRelationships~LegalEmployerName	legalEmployerNameWithLegislationCode	Legal EmployerName with Legislation Code	HCM
workRelationships~TerminationDate	terminationDate	Termination Date	HCM
workRelationships~PeriodOfServiceId	periodOfServiceId	PeriodOfService Id	HCM
workRelationships~LegalEntityId	legalEntityId	Legal Entity Id	HCM
workRelationships~assignments~EffectiveStartDate	assignmentEffectiveStartDate	Assignment Effective Start Date	HCM
workRelationships~assignments~PositionCode	positionCode	Position Code	HCM
workRelationships~assignments~GradeCode	gradeCode	Grade Code	HCM
workRelationships~assignments~LocationCode	locationCode	Location code	HCM
workRelationships~assignments~EffectiveEndDate	assignmentEffectiveEndDate	Assignment Effective End Date	HCM
workRelationships~assignments~ActionCode	actionCode	Action Code	HCM
workRelationships~assignments~ActionTypeCode	actionTypeCode	Action Type Code	HCM
workRelationships~assignments~ProjectedStartDate	projectedStartDate	ProjectedStartDate	HCM
workRelationships~assignments~ProposedUserPersonType	proposedUserPersonType	Proposed User Person Type	HCM
workRelationships~assignments~managers~ManagerAssignmentNumber	managerID	Manager Identification Number	HCM
addresses~Country	country	Location	HCM
addresses~AddressType	addressType	Address Type	HCM
addresses~PostalCode	postalCode	Location postal code	HCM
addresses~TownOrCity	townOrCity	Town or city	HCM
addresses~Region1	region1	Region1	HCM
addresses~Region2	region2	Region2	HCM
addresses~FloorNumber	floorNumber	Floor number	HCM
addresses~Building	building	Building	HCM
addresses~AddressLine1	addressLine1	AddressLine1	HCM

Table 5-29 (Cont.) Default Attributes - Authoritative Source for HCM with JML Support

Attribute Name on Oracle Fusion Cloud Applications	Attribute Name on Oracle Access Governance	Display Name on Oracle Access Governance	Applicable to HCM/ ERP/ BOTH
phones~PhoneNumber	phoneNumber	Phone Number	HCM
phones~Extension	extension	Phone Extension	HCM
phones~PhoneType	phoneType	Phone Type	HCM
Work Relationships	workRelationships	Work Relationships	HCM
			HCM
			HCM
			HCM

Table 5-30 Default Attributes - Managed System

Oracle Fusion Cloud Applications User Entity	Oracle Fusion Cloud Applications Attribute Name	Display Name on Oracle Access Governance	Oracle Access Governance Attribute Display Name	Applicable to HCM/ ERP/ BOTH
FA User	id (SCIM)	uid	Unique Id	Both
	userName	name	Name	Both
	password	password	Password	Non-Reconcilable
	externalId	externalID	External ID	Both
	displayName	displayName	Display Name	Both
	name.familyName	familyName	Family Name	Both
	name.givenName	givenName	Given Name	Both
	emails.value	email	Email	Both
	emails.type	emailType	Email Type	Both
	active	status	Status	Both
	workerInformation.personNumber	personNumber	Person Number	Both
	workerInformation.manager	manager	Manager	Both
	workerInformation.job	jobCode	Job Code	Both
Roles	securityContextsWithValues	Roles	Roles	Both ERP

Oracle Health EHR (formerly Cerner Millenium)

Overview: Integrate Oracle Access Governance with Oracle Health EHR (formerly Cerner Millenium)

You can integrate Oracle Access Governance with Oracle Health EHR (formerly Cerner Millenium) for enabling identity orchestration, including on-boarding of identity user data and provisioning of Oracle Cerner accounts.

You can establish a connection between Oracle Health EHR (formerly Cerner Millenium) and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance.

Oracle Health EHR (formerly Cerner Millenium) Integration Architecture Overview

The integration of Oracle Health EHR (formerly Cerner Millenium) allows for retrieving identity data and transferring the data to Oracle Access Governance.

Oracle Health EHR (formerly Cerner Millenium) integration is implemented using an Agent-based connection type. This means that a direct connection is not available, so an indirect connection is made between Oracle Health EHR (formerly Cerner Millenium) and the required Cerner Millennium instance using the Access Governance Agent.

The Oracle Health EHR (formerly Cerner Millenium) application communicates with the Cerner API using the HTTP protocol. The Cerner API provides programmatic access to Cerner through the endpoint. Oracle Health EHR (formerly Cerner Millenium) applications uses the endpoints to perform create, read, and update, operations on directory data and directory objects, such as users, personnel groups, Organization, Organization Groups and Personal alias.

Oracle Health EHR (formerly Cerner Millenium) Integration Functional Overview

Oracle Health EHR (formerly Cerner Millenium) integration supports configuration of the Cerner accounts which include user account creation, update, change password, and assigning and removal of roles.

Configure Oracle Health EHR (formerly Cerner Millenium) Orchestrated System

The first task you need to carry out is to set up and configure Oracle Health EHR (formerly Cerner Millenium) Orchestrated System. This gives Oracle Access Governance the details for how to connect to the Oracle Health EHR (formerly Cerner Millenium) system from which you want to load data, or manage permissions. Optionally you can configure further elements of the Orchestrated System before running the initial dataload including:

- Notification Settings
- Identity/Account Matching Rules
- Apply data transformations to inbound and outbound data
- Identity attributes

Load Data

After setting up and verifying your Orchestrated System, you can ingest identity and account details from Oracle Health EHR (formerly Cerner Millenium), using the configuration mode - Managed System.

User data loaded in Managed System mode comprises of account data and roles of Oracle Health EHR (formerly Cerner Millenium). If the account is new, then a new account is created in Oracle Access Governance together with the associated roles, These roles will be created in Oracle Access Governance as permissions. Accounts and permissions loaded from Oracle Health EHR (formerly Cerner Millenium) can be managed by Oracle Access Governance. You can update the permissions associated with a managed system account. If the account only has one permission assigned then remediation of this permission will also result in the revoking of the account. If the user details such as identities exists in , then the updates initiated using system is applied.

Create Account

As an Oracle Access Governance user you can request access to resources and roles provided in Request Access .

The following ways allows you to create an user account in Oracle Access Governance:

- Ingestion of user records as data from Oracle Health EHR (formerly Cerner Millenium).
- When a role, policy, or access bundle containing Oracle Health EHR (formerly Cerner Millenium) roles is assigned to an identity. If you have an identity in Oracle Access Governance then you can request an account by using the **Request a new access** functionality in the Oracle Access Governance console. If you make an access request for an access bundle, or role, after approval, a provisioning operation is initiated. The provisioning process will, if there is not an existing account managed by Oracle Access Governance, create an account on the Oracle Health EHR (formerly Cerner Millenium) instance. If an account managed by Oracle Access Governance already exists, then the Oracle Health EHR (formerly Cerner Millenium) roles for that account are updated based on the values in the access bundle.

Assign Permissions

You can assign permissions to a Oracle Health EHR (formerly Cerner Millenium) account using the **Request a new access** functionality of Oracle Access Governance. This allows you to request an access bundle containing permissions which equate to roles on the Oracle Health EHR (formerly Cerner Millenium) system. When you request an access bundle, either directly or through an Oracle Access Governance role or policy, a provisioning operation is initiated which updates the roles in your Oracle Health EHR (formerly Cerner Millenium) instance with the permissions included in the referenced access bundle.

For further details about permission assignment, refer to Request Access. To learn more about roles and policies, refer to Manage Roles, and Manage Policies.

Remove Permissions

You can remove permissions from an account by revoking the permissions from the role, policy or access bundle to which it is assigned. In this case, the permission assignment is revoked from all users to whom the role, policy or access bundle is applied. Another way to remove a permission would be by revoking role, policy or access bundle assignment from a specific account. This would be done using the revoke operation in access reviews.

For further details about permission assignment, refer to Delete a Role, Delete a Policy, or Manage Access Bundles -> Delete an Access Bundle.

Change Password

The ability to change an account password is provided by the **My Access** functionality in Oracle Access Governance Console. If you change the account password in this page, the details will be sent to the Oracle Health EHR (formerly Cerner Millenium) instance in the next provisioning operation.

For more details, refer to Change Account Password.

Oracle Health EHR (formerly Cerner Millenium) Components Certified for Integration with Oracle Access Governance

The Oracle Health EHR (formerly Cerner Millenium) to integrate with Orchestrated System is as follows.

Certified Components

Table 5-31 Certified Components

Component Type	Component
System	The versions of Oracle Health EHR (formerly Cerner Millenium) you can use for Oracle Access Governance are: • Cerner Millennium, Security provisioning engine version 7.0.0

Supported Configuration Modes for Oracle Health EHR (formerly Cerner Millenium)

You can use Oracle Access Governance integrations to set up different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Supported Modes

The Oracle Health EHR (formerly Cerner Millenium) Orchestrated System supports the following modes:

- **Managed System**
  You can manage Oracle Health EHR (formerly Cerner Millenium) user profile records.

Supported Operations When Provisioning To Oracle Health EHR (formerly Cerner Millenium)

To provision an account from Oracle Access Governance to Oracle Health EHR (formerly Cerner Millenium) there are certain operations that are supported.

The Oracle Health EHR (formerly Cerner Millenium) Orchestrated System supports the following account operations when provisioning a user:

- Create Account
- Update Account
- Enable Account
- Disable Account
- Password Change
- Add Organization
- Remove Organization
- Add Organization Group
- Remove Organization Group
- Add Personal Alias

- Remove Personal Alias
- Add Personal Group
- Remove Personal Group
- Add PPR access
- Remove PPR access
- Add Taxonomy
- Remove Taxonomy
- Add Role profile
- Remove Role profile

Default Supported Attributes

Oracle Access Governance supports the following default Oracle Health EHR (formerly Cerner Millenium) attributes.

These attributes are mapped depending on the direction of the connection, for example:

- Data for user accounts, and personal alias as Identities being ingested by Oracle Access Governance from Oracle Health EHR (formerly Cerner Millenium).
- Data for the user accounts as Identities being ingested by Oracle Access Governance from Oracle Health EHR (formerly Cerner Millenium).

Table 5-32 Default Attributes

Oracle Health EHR (formerly Cerner Millenium) User Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name
	ID	uid	Unique Id
	username	name	User login
	firstName	firstName	First name
	lastName	lastName	Last name
	middleName	middleName	Middle name
	suffix	suffix	Honorific suffix
	title	title	Title
	PhysicianInd	physician	Physician
	gender	gender	Gender
	directoryIndicator	directory	Directory
	logicalDomain	logicalDomain	Logical domain
	birthdate	dateOfBirth	Date of birth
	beginEffectiveDateTime	startDate	Start date
	endEffectiveDateTime	endDate	End date
	position	position	Position
	inactiveAccount	status	Status
	Password	password	Password
Organization	Organization	organizations	Organizations
PersonalGroup	personnelGroup	personalGroups	Personal groups
Organization Group	organizationGroup	organizationGroups	Organization groups
Personal alias	personnelAlias	personalAliases	Personal aliases

Table 5-32 (Cont.) Default Attributes

Oracle Health EHR (formerly Cerner Millenium) User Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name
	toPsoID	aliasId	Alias id
	aliasPool	aliasPool	Alias pool
	type	aliasType	Alias type
	alias	aliasName	Alias name
	beginEffectiveDateTime	startDate	Start date
	endEffectiveDateTime	endDate	End date
Patient Provider Relationship (PPR access)	pprAccess	pprAccesses	PPR acesses
Taxonomy	Taxonomy	taxonomies	Taxonomies
	toPsoID	taxonomyPsoId	Taxonomy PSO id
	taxonomyId	taxonomyId	Taxonomy id
	taxonomyType	taxonomyType	Taxonomy type
	beginEffectiveDateTime	startDate	Start date
	endEffectiveDateTime	endDate	End date
Role Profile	roleProfile		
	toPsoID	toPsoID	Role PDO id
	position	position	Position
	display	display	display

Troubleshooting

The following are the troubleshooting steps for Oracle Health EHR (formerly Cerner Millenium).

- The following error message is displayed while performing operations:
  ERROR

  Error occurred while establishing SOAP connection "OR' call: Connection Refused: connect.

  Resolution

  The Cerner server must be running and accessible from the agent location.
- Currently only addition of alias is supported, you cannot perform **Removal of Alias** operation.

PeopleSoft

Overview: Integrate Oracle Access Governance with PeopleSoft

Oracle Access Governance can be integrated with PeopleSoft enabling identity orchestration, including on-boarding of identity (user) data, and provisioning of PeopleSoft accounts.

PeopleSoft provides enterprise Human Capital Management (HCM) and Enterprise Resource Planning (ERP) functionality. Oracle Access Governance supports the following elements within PeopleSoft:

- PeopleSoft HRMS as an authoritative (trusted) source of identity information allowing for reconciliation of employees created or modified in PeopleSoft HRMS.
- PeopleSoft User Management as a Managed System enabling provisioning of PeopleSoft application accounts.

Peoplesoft Integration Architecture Overview

Integration with PeopleSoft allows you to retrieve identity data from a system, transport it to Oracle Access Governance, and ingest. Once a system is connected, you can perform provisioning and remediation tasks which are then reflected in the Managed System.

PeopleSoft integration is implemented using an Agent-based connection type. This means that a direct connection is not available, so an indirect connection is made between Oracle Access Governance and the required PeopleSoft instance using the Access Governance Agent. The PeopleSoft integration supports the following flows:

- If you select the Authoritative Source configuration mode when you setup a PeopleSoft Orchestrated System, then Oracle Access Governance will retrieve identity data from the PeopleSoft instance and treat it as an authoritative (trusted) source of identity information.
- If you select the Managed Systems configuration mode, then Oracle Access Governance will allow you to manage PeopleTools-based PSOPRDEFN user profile records in PeopleSoft applications. This enables the provisioning of new accounts in PeopleSoft from Oracle Access Governance.

The connection is made through PeopleSoft's Component Interface. This results in a full load of relevant identity and account data into Oracle Access Governance each time the load is executed. If this is the first time that the load is made, then relevant identity and account structures are created in Oracle Access Governance as appropriate. On subsequent dataload runs, all data is loaded to Oracle Access Governance and the ingestion process updates any changes since the last dataload in the appropriate identity and account artefacts.

Once the connection and Day0 dataload are completed, you can provision accounts using Oracle Access Governance's provisioning engine which will take any provisioning request and pass it through the agent and onwards to PeopleSoft. Provisioning supports create, update, and revoke operations.

PeopleSoft Integration Functional Overview

PeopleSoft integration supports usecases for HRMS and ERP including configuration of the Orchestrated System, dataload, account creation and revocation, change password, and assign and remove roles.

Configure PeopleSoft Orchestrated System

The first task you need to carry out is setup and configuration of the PeopleSoft Orchestrated System. This gives Oracle Access Governance the details for how to connect to the PeopleSoft system from which you want to load data, or manage permissions. Optionally you can configure further elements of the Orchestrated System before running the initial dataload including:

- Notification Settings
- Identity/Account Matching Rules

- Apply data transformations to inbound and outbound data
- Identity attributes

Load Data

Once you have setup and verified your Orchestrated System, you can run dataloads to ingest identity and account details from PeopleSoft, depending on the configuration mode you have selected, *Authoritative Source* or *Managed System*.

Data loaded in Authoritative Source mode will consist of user data from the PeopleSoft system. If the user is new, then a new identity is created in Oracle Access Governance. If the identity already exists in Oracle Access Governance, then any updates initiated in the PeopleSoft system will be applied.

Data loaded in Managed System mode comprises account data and roles from PeopleSoft. If the account is new, then a new account is created in Oracle Access Governance together with the associated roles, These roles will be created in Oracle Access Governance as permissions. Accounts and permissions loaded from PeopleSoft can be managed by Oracle Access Governance. You can remediate permissions associated with a managed system account. If the account only has one permission assigned then remediation of this permission will also result in the revoking of the account.

Create Account

An account can be created in Oracle Access Governance in two ways:

- Ingested account data from PeopleSoft.
- When a role, policy, or access bundle containing PeopleSoft roles is assigned to an identity. If you have an identity in Oracle Access Governance then you can request an account by using the **Request a new access** functionality in the Oracle Access Governance console. If you make an access request for an access bundle, or role, once approved, a provisioning operation will be initiated. The provisioning process will, if there is not an existing account managed by Oracle Access Governance, create an account on the PeopleSoft instance. If an account managed by Oracle Access Governance already exists, then the PeopleSoft roles for that account are updated based on the values in the access bundle.

The account created in PeopleSoft equates to a PeopleTools-based PSOPRDEFN user profile record.

For further details about account creation, refer to Request Access.

Assign Permissions

You can assign permissions to a PeopleSoft account using the **Request a new access** functionality of Oracle Access Governance. This allows you to request an access bundle containing permissions which equate to roles on the PeopleSoft system. When you request an access bundle, either directly or through an Oracle Access Governance role or policy, a provisioning operation is initiated which updates the roles in your PeopleSoft instance with the permissions included in the referenced access bundle.

For further details about permission assignment, refer to Request Access. To learn more about roles and policies, refer to Manage Roles, and Manage Policies.

Remove Permissions

You can remove permissions from an account by revoking the permissions from the role, policy or access bundle to which it is assigned. In this case, the permission assignment is revoked from all users to whom the role, policy or access bundle is applied. Say you had an access

bundle with two permissions, *PSFT_Admin*, and *PSFT_Developer* which had previously been provisioned to PeopleSoft, you could update the access bundle containing these permissions to remove PSFT_Developer and add *PSFT_Composer*, resulting in the access bundle containing PSFT_admin, and PSFT_Composer. This change would be reflected following the next provisioning operation by removing the PSFT_Developer role and assigning the PSFT_Composer role. PSFT_Admin would remain assigned.

Another way to remove a permission would be by revoking role, policy or access bundle assignment from a specific account. This would be done using the revoke operation in access reviews.

For further details about permission assignment, refer to Delete a Role, Delete a Policy, or Manage Access Bundles -> Delete an Access Bundle.

Change Password

The ability to change an account password is provided by the **My Access** functionality in Oracle Access Governance Console. If you change the account password in this page, the details will be sent to the PeopleSoft instance in the next provisioning operation, and the password change is applied you your PeopleSoft account.

For further details about changing passwords, refer to Change Account Password.

Revoke Account

If you select to revoke an account within an access review, provisioning tasks will be created to revoke the account within PeopleSoft. For further details about revoking accounts, refer to Delete a Role, or Delete a Policy.

An Example Account Lifecycle

Let's look at an example. You have created a new Orchestrated System which is connected to the **MyPSFT** instance which contains HRMS and ERP data for your organization. The Orchestrated System is configured for Authoritative Source and Managed System modes. On the first dataload, identity and account data is loaded into Oracle Access Governance. At this time the following details are created in Oracle Access Governance:

- An Oracle Access Governance identity is created, say*MyAGIdentity*, comprising authoritative data such as name, email, and location.
- An account is created in Oracle Access Governance for existing PeopleSoft roles, say *PSFTRole_Composer*.

We now have the following:

- **MyAGIdentity**
  - MyPSFTAccount
    - PSFTRole_Composer

After some time *MyAGIdentity* moves into a development role requiring the PeopleSoft developer role. An access bundle *PSFTBundle_Developer* is created in Oracle Access Governance which contains the development permissions required. This access bundle can be assigned as a result of a policy, role or request. Let's say the user requests the access bundle using the **Request a new access** option. On approval, the request triggers a provisioning operation which applies the changes to **MyPSFT**, assigning the PeopleSoft roles corresponding to the permissions contained in *PSFTBundle_Developer* access bundle.

We now have the following:

- **MyAGIdentity**

    - MyPSFTAccount
        * PSFTRole_Composer
        * PSFTBundle_Developer

Additional accounts may be mapped to the *MyAGIdentity* identityover time from other Managed Systems giving us a profile like this:

- **MyAGIdentity**
    - MyPSFTAccount
    - MyOracleDBAccount
    - MyMSTeamsAccount

MyAGIdentity then decides to change his password. Using the **My Access** functionality in Oracle Access Governance Console, he changes his password, which propagates the change to **MyPSFT** using Oracle Access Governance provisioning.

MyAGIdentity then moves into a role which means they no longer require an account on PeopleSoft. In this case a revoke account provisioning task can be generated by revoking the identity's account as part of an access review. Alternatively, their association with PeopleSoft roles can be removed by removing the identity from the relevant Oracle Access Governance role or policy. In either case, this will result in a provisioning task which will revoke the account from PeopleSoft, together with any related roles. The profile would now resemble:

- **MyAGIdentity**
    - MyOracleDBAccount
    - MyMSTeamsAccount

If the PeopleSoft Orchestrated System is configured in *Authoritative Source* mode and you make an identity inactive then the the *MyAGIdentity* identity, is effectively disabled. In this case a provisioning task will be generated and provisiong to the Managed System.

We now have the following:

- **MyAGIdentity (Disabled)**

Prerequisites

Before you install and configure a PeopleSoft Orchestrated System, you should consider the following prerequisites and tasks.

1. Your PeopleSoft instance is certified with Oracle Access Governance. Refer to Peoplesoft Components Certified for Integration with Oracle Access Governance for details of the versions supported.
2. Your environment meets the requirements for certain PeopleSoft elements to be present in your environment. See PeopleSoft Components Required For Integration for details of the requirements.

Configure

You can establish a connection between PeopleSoft and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon, select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select System

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field.

1. Select **PeopleSoft**.
2. Click **Next**.

Enter Details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.

2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account Settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration Settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to PeopleSoft.

1. Enter the **Easy Connect URL for database** for connection to the PeopleSoft Oracle database.
2. In the **Database schema user name** enter the name of the schema user you will use to connect to the PeopleSoft Oracle database. For details on how to create this user, refer to Configure Oracle Database Schema User Account.
3. Enter the **Database schema password** for the schema user you will use to connect to the PeopleSoft Oracle database. Confirm your password in the **Confirm password** field.
4. In the **Url** field, enter the URL of the server hosting the PeopleSoft application server you want to integrate with. Use the format `host:port`, where `port` is the PeopleSoft Jolt port.
5. In the **Username** field, enter the username required to connect to the PeopleSoft instance to perform data reconciliation and provisioning. For details on how to create this user, refer to Configure PeopleSoft Service Account Using Peoplesoft PIA Web Interface.
6. In the **Password** and **Confirm password** fields enter the password that authenticates the user you are connecting to the PeopleSoft instance with.
7. Optionally, if you have domains configured in your PeopleSoft instance, enter the PeopleSoft domain password into the **Domain password** and **Confirm password** fields.
8. In the **Custom jar details** field enter the details of custom PeopleSoft jars used during integration. The files are `psjoa.jar` and `psmanagement.jar`, and should be entered in the following format:

   ```
   <jarName>::<jarChecksum>, <jarName>::<jarChecksum>
   ```

   For example:

   ```
   psjoa.jar::12345, psmanagement.jar::54321
   ```

   For more information on custom jar support, refer to Custom Jar Support.
9. Click **Add** to create the orchestrated system.

Finish Up

The final step of the workflow is **Finish Up** where you are prompted to download the agent for your Orchestrated System. Once you have downloaded the agent, you can install and configure the agent in your environment using the instructions in Install Oracle Access Governance Agent.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

When you have complete installation of your agent, you need to copy two Java jar files from the PeopleSoft installation, into the agent custom JARs directory.

1. Copy the following JAR files from your PeopleSoft home directory (PS_HOME).
   - psjoa.jar
   - psmanagement.jar

2. Copy the files to the agent custom JARs directory path as specified in the `config.json`, where `/app` is the agent volume. For example, you may have a value such as `"customJarsDirectory": "/app/data/customJars"` in your `config.json`.

> **Note:**
>
> If the custom JARs are not available for any reason, then the Validate operation will fail with the error message **Invalid session**.

Components Certified for Integration with Oracle Access Governance

The components that you can integrate with, depend on which configuration mode you are running your Orchestrated System in.

Certified Components in Authoritative Source Configuration Mode

Table 5-33 Certified Components in Authoritative Source Configuration Mode

Component Type	Component
System	The versions of HRMS you can use as an authoritative (trusted) source of identity information for Oracle Access Governance are: • PeopleSoft HRMS 8.9 with PeopleTools 8.49 • PeopleSoft HRMS 8.9 with PeopleTools 8.50 • PeopleSoft HRMS 9.0 with PeopleTools 8.49 • PeopleSoft HRMS 9.0 with PeopleTools 8.50 • PeopleSoft HRMS 9.0 with PeopleTools 8.52 • PeopleSoft HRMS 9.1 with PeopleTools 8.50 • PeopleSoft HRMS 9.1 with PeopleTools 8.51 • PeopleSoft HRMS 9.1 with PeopleTools 8.52 • PeopleSoft HRMS 9.1 with PeopleTools 8.53 • PeopleSoft HRMS 9.2 with PeopleTools 8.53 • PeopleSoft HRMS 9.2 with PeopleTools 8.54 • PeopleSoft HRMS 9.2 with PeopleTools 8.55 • PeopleSoft HRMS 9.2 with PeopleTools 8.56 • PeopleSoft HRMS 9.2 with PeopleTools 8.57 • PeopleSoft HRMS 9.2 with PeopleTools 8.58 • PeopleSoft HRMS 9.2 with PeopleTools 8.59

Certified Components in Managed System Configuration Mode

Table 5-34 Certified Components in Managed System Configuration Mode

Component Type	Component
System	The versions of PeopleTools you can use to manage PeopleTools-based PSOPRDEFN user profile records in PeopleSoft applications are: • PeopleTools 8.53 • PeopleTools 8.54 • PeopleTools 8.55 • PeopleTools 8.56 • PeopleTools 8.57 • PeopleTools 8.58 • PeopleTools 8.59 • PeopleTools 8.60.05 • PeopleTools 8.61.03 **Note:** If you are using PeopleTools 8.54, full reconciliation operation may not work as expected. Apply PeopleSoft Patch 21109998 using the following URL for this operation to work successfully: `https://support.oracle.com/`

Certified Components in both Modes

Table 5-35 Certified Components in both Modes

Component Type	Component
System	The versions of HCM you can use in either Authoritative Source or Managed System mode are: • PeopleSoft HCM 9.1 • PeopleSoft HCM 9.1
Database	Oracle

Components Required For Integration with Oracle Access Governance

Integration of with Oracle Access Governance requires a number of components to be present in your environment.

Ensure the following components are installed in your environment:

- Tuxedo and Jolt (the application server)
- Internet Architecture (PIA)
- Application Designer (2-tier mode)

Configure Oracle Database Schema User Account

To access the database schema you will need to create a service account on the Oracle database supporting .

1. Log in to the Oracle database as a database administrator using SQL*Plus or similar. Create a service account using the following statements:

   ```
   create user <DBService Schema user account name> identified by <password>;
   grant create session to <DBService Schema user account name>;
   grant create synonym to <DBService Schema user account name>;
   ```

   For example:

   ```
   create user psftagsvc identified by mypw;
   grant create session to psftagsvc;
   grant create synonym to psftagsvc;
   ```

2. Grant permission to schema components to the service account you created, where `<PSFT>` is the name of the schema for your environment:

   ```
   grant select on <PSFT>.PSOPRDEFN to <DBService Schema user account name>;
   grant select on <PSFT>.PSROLEDEFN_SRCH to <DBService Schema user account
   name>;
   grant select on  <PSFT>.PSCLASSDEFN to <DBService Schema user account
   name>;
   grant select on  <PSFT>.PS_CURRENCY_CD_TBL to <DBService Schema user
   account name>;
   grant select on  <PSFT>.PS_PERSONAL_DATA to <DBService Schema user account
   name>;
   grant select on  <PSFT>.PS_PERSONAL_PHONE to <DBService Schema user
   account name>;
   grant select on  <PSFT>.PS_EMAIL_ADDRESSES to <DBService Schema user
   account name>;
   grant select on  <PSFT>.PS_JOB to <DBService Schema user account name>;
   grant select on  <PSFT>.PS_JOBCODE_TBL to <DBService Schema user account
   name>;
   ```

3. Logout of the database and reconnect as the service account you created. Create synonyms for the schema components you granted access for:

   ```
   create synonym PSOPRDEFN for <PSFT>.PSOPRDEFN;
   create synonym PSROLEDEFN_SRCH for <PSFT>.PSROLEDEFN_SRCH;
   ```

```
create synonym PSCLASSDEFN for <PSFT>.PSCLASSDEFN;
create synonym CURRENCY_CD_TBL for <PSFT>.PS_CURRENCY_CD_TBL;
create synonym PS_PERSONAL_DATA for <PSFT>.PS_PERSONAL_DATA;
create synonym PS_PERSONAL_PHONE for <PSFT>.PS_PERSONAL_PHONE;
create synonym PS_EMAIL_ADDRESSES for <PSFT>.PS_EMAIL_ADDRESSES;
create synonym PS_JOB for <PSFT>.PS_JOB;
create synonym PS_JOBCODE_TBL for <PSFT>.PS_JOBCODE_TBL;
```

Configure Service Account Using Peoplesoft PIA Web Interface

Integrating with requires connecting to the application using a service account.

You can create a service user to connect to the application with, by executing the following steps.

1. Invoke the **Peoplesoft PIA Web interface** in a browser and navigate to Permission Lists. **People Tools → Security → Permission Lists**
2. Add a new value: AGCS_PERMLIST
3. In the permission list add and assign access to the following Component Interfaces according to the values given in the table:

Table 5-36 Component Interface Permissions

Component Interface	Method	Method Access
USER_PROFILE		
	Cancel	Full Access
	Get	Full Access
	Create	Full Access
	Save	Full Access
	ResetPassword	Full Access
	ResetPassword_Alpha	Full Access
	SetPassword	Full Access
	SetDescription	Full Access
DELETE_USER_PROFILE		
	Cancel	Full Access
	Find	Full Access
	Get	Full Access
	Save	Full Access
ROLE_MAINT		
	Cancel	Full Access
	Find	Full Access
	Get	Full Access
	Create	No Access
	Save	No Access
CURRENCY		
	Cancel	Full Access
	Find	Full Access
	Get	Full Access
	Create	No Access

Table 5-36 (Cont.) Component Interface Permissions

Component Interface	Method	Method Access
	Save	No Access
CI_PERM_LIST		
	Cancel	Full Access
	Find	Full Access
	Get	Full Access
	Create	No Access
	Save	No Access

4. Navigate to Roles.
   People Tools → Security → Roles
5. Add a new value: AGCS_ROLE
6. Add AGCS_PERMLIST to the Permission List.
7. Navigate to User Profile
   People Tools → Security → User Profiles → User Profile
8. Add a new value: AGCSSA
   - Add Symbolic ID as SYSADM1.
   - Set and confirm the password.
   - Set ID Type as NONE.
   - From Roles, select AGCS_ROLE.
   - Save your changes.

Supported Configuration Modes for Integrations

Oracle Access Governance integrations can be setup in different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Supported Modes

Orchestrated System supports the following modes:

- **Authoritative Source**
  You can use HRMS as an authoritative (trusted) source of identity information for Oracle Access Governance.
- **Managed System**
  You can manage PeopleTools-based PSOPRDEFN user profile records in applications including Role and Permission List assignments to these records.

Supported Operations When Provisioning To

When you provision an account from Oracle Access Governance to certain operations are supported.

The Orchestrated System supports the following account operations when provisioning a user:

- Create User
- Update User
- Change Password

- Add Roles
- Remove Roles

Default Supported Attributes

Oracle Access Governance supports the following default attributes. These attributes are mapped depending on the direction of the connection, for example:

- Data being ingested by Oracle Access Governance from : `User.PROP_FIRST_NAME` will map to `Identity.firstName`
- Data being provisioned into from Oracle Access Governance: `account.lastName` will map to `User.PROP_LAST_NAME`

Table 5-37 Default Attributes - Authoritative Source

Entity	Attribute Name On Managed System	Oracle Access Governance Identity Attribute Name	Oracle Access Governance Identity Attribute Display Name
User	UserID	uid	Unique Id
	UserID	name	Employee user name
	IDTypes~EMP~Empl_ID	employeeNumber	Employee number
	IDTypes~CST~Set_ID	customerSetId	Customer set id
	IDTypes~CST~Customer_ID	customerId	Customer id
	IDTypes~VND~Set_ID	vendorSetId	Vendor set id
	IDTypes~VND~Vendor_ID	vendorId	Vendor id
	PROP_FIRST_NAME	firstName	First name
	PROP_LAST_NAME	lastName	Last name
	PROP_MIDDLE_NAME	middleName	Middle name
	PROP_NAME_TITLE	title	Title name
	PROP_EMAIL_ADDR	email	Email
	PROP_PHONE	phone	Phone
	PROP_DEPTID	department	Department
	PROP_JOBCODE	jobCode	Job code
	PROP_POSITION_NBR	positionNBR	Position
	PROP_SUPERVISOR_ID	supervisorUid	Supervisor
	PROP_HR_STATUS	hrStatus	HR status
	PROP_EMPL_STATUS	emplStatus	Employee Status
	PROP_ACTION	action	Action
	PROP_ACTION_REASON	actionReason	Action reason
	PROP_LOCATION	location	Location
	PROP_FULL_PART_TIME	fullPartTime	Full part time
	PROP_COMPANY	company	Company
	PROP_EMPL_TYPE	emplType	Employee type
	PROP_EMPL_CLASS	emplClass	Employee class

Table 5-37 (Cont.) Default Attributes - Authoritative Source

Entity	Attribute Name On Managed System	Oracle Access Governance Identity Attribute Name	Oracle Access Governance Identity Attribute Display Name
	PROP_BUSINESS_UNIT	businessUnit	Business unit
	PROP_TERMINATION_DT	endDate	End date
	EFFDT	startDate	Start date
	REPORTS_TO	reportsTo	Reports to
	PROP_OFFICER_CD	officerCode	Officer code
addresses		addresses as entitlement	Addresses
	PROP_COUNTRY	country	
	PROP_CITY	city	
	PROP_STATE	state	
	PROP_ADDRESS1	address1	
	PROP_ADDRESS2	address2	
	PROP_ADDRESS3	address3	
	PROP_POSTAL	postal	

Table 5-38 Default Attributes - Managed System

Entity	Attribute Name On Managed System	Oracle Access Governance Account Attribute Name	Oracle Access Governance Account Attribute Display Name
User	UserID	uid	Unique Id
	UserID	name	User login
	__PASSWORD__	password	Password
	EmailAddresses~EmailAddress~PrimaryEmail	email	Email
	IDTypes~EMP~Empl_ID	employeeId	Employee id
	IDTypes~CST~Set_ID	customerSetId	Customer set id
	IDTypes~CST~Customer_ID	customerId	Customer id
	IDTypes~VND~Set_ID	vendorSetId	Vendor set id
	IDTypes~VND~Vendor_ID	vendorId	Vendor id
	NavigatorHomePermissionList	navigatorHomePermission	Navigator home permission
	ProcessProfilePermissionList	processProfilePermission	Process profile permission
	RowSecurityPermissionList	rowSecurityPermission	Row security permission
	PrimaryPermissionList	primaryPermission	Primary permission
	UserDescription	description	Description
	MultiLanguageEnabled	multiLanguageEnabled	Multi language enabled
	SymbolicID	symbolicId	Symbolic id
	UserIDAlias	userIdAlias	User id alias

Table 5-38 (Cont.) Default Attributes - Managed System

Entity	Attribute Name On Managed System	Oracle Access Governance Account Attribute Name	Oracle Access Governance Account Attribute Display Name
	LanguageCode	languageCode	Language
	CurrencyCode	currencyCode	Currency
	AlternateUserID	alternateUserId	Alternate user id
	EffectiveDateFrom	startDate	Start date
	EffectiveDateTo	endDate	End date
	WorklistUser	worklistUser	Work list user
	EmailUser	emailUser	Email user
	ReassignWork	reassignWork	Reassign work
	ReassignUserID	reassignUserId	Reassign work to
	SupervisingUserID	supervisingUserId	Supervising user id
	AccountLocked	status	Status

Default Matching Rules

In order to map accounts to identities in Oracle Access Governance you need to have a matching rule for each Orchestrated System.

The default matching rule for orchestrated system is:

Table 5-39 Default Matching Rules

Mode	Default Matching Rule

Authoritative Source
Identity matching checks if incoming identities match an existing identity or are new
	Screen value:
	`Employee user name = Employee user name`
	Attribute name:
	`Identity.userName = Identity.userName`

Managed System
Account matching checks if incoming accounts match with existing identities.
	Screen value:
	`User login = Employee user name`
	Attribute name:
	`Account.name = Identity.userName`

SAP Ariba

Integrate Oracle Access Governance with

Oracle Access Governance enables **API-based** seamless integration with for enabling identity orchestration, automating onboarding of accounts and groups, provisioning and reconciliation of accounts. Oracle Access Governance supports account management and group management for accounts as a **Managed System**.

SAP Ariba is a comprehensive cloud-based procurement and spend management service that helps businesses streamline and optimize their procurement processes, from sourcing to payment. With this integration, you can create, update, enable, and disable identity accounts. You can assign or revoke groups for accounts from Oracle Access Governance.

Overview: Orchestrated System

You can establish a connection between and Oracle Access Governance by entering connection details and configuring the orchestrated system. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

Integration Architecture Overview

You can perform full data and incremental data load for identities in . Once a connection is established, you can perform provisioning and remediation tasks for user accounts and groups.

integration leverages two APIs for provisioning and full and/or incremental data load.

- Use the **Import Users** SOAP API to create, update, enable, disable accounts to existing supplier or customer organizations for SAP Ariba Strategic Sourcing solutions. You can assign and/or revoke identities from the groups as from Oracle Access Governance.
- Use the **Master Data Retrieval API for Sourcing** REST API to read data, and perform full or incremental data load in Oracle Access Governance.

Functional Overview: Use Cases Supported for Integration

integration supports account management and group management for accounts. The SAP Ariba orchestrated system supports management of accounts for *Cloud Identity*, *Synchronized Identity*, and *Federated Identity* models of .

Configure Orchestrated System

First, set up and configure Orchestrated System. For details, see Configure Integration Between Oracle Access Governance and SAP Ariba. This configuration provides Oracle Access Governance the connection details on how to load data and manage permissions for this Orchestrated System.

Load Data

After setting up and verifying your Orchestrated System, you can ingest account details from , using the configuration mode - **Managed System**. This indicates that accounts and groups are ingested into Oracle Access Governance and can be managed by Oracle Access Governance.

Users in are ingested as accounts and groups are ingested as permissions in Oracle Access Governance.

Account Management for - Create, Update, Enable and Disable Account

The following ways allows you to create an account in Oracle Access Governance:

- Ingestion of account and permissions as part of data load operation from .
- When a role, policy, or access bundle containing groups are assigned to an identity. Any active identity in Oracle Access Governance can request permissions by using the self-service **Request a new access** functionality in the Oracle Access Governance console. If you make an access request for an access bundle, or role, then after approval, a provisioning operation is initiated.

Here's how you can manage account operations using Oracle Access Governance

- **Create Account**: New accounts in are created as part of group assignment. If you request groups for an identity that does not have corresponding account in , a new account is created and the requested groups based on the access bundle values, are assigned to it.

To associate new accounts and groups, the orchestrated system triggers **Create Account** and **Add Child Data** operations.

- **Update Account**: If an account is managed by Oracle Access Governance and corresponding account already exists in , then groups for that account are updated based on the values in the access bundle. **Update Account** provisioning task is triggered along with **Remove Child Data** and **Add Child Data**
- **Enable Account**: If only permissions are different, then account remains enabled but **Add Child Data** and/or **Remove Child Data** operations are triggered in the Orchestrated System to update the permissions for that account.
- **Disable Account**: If all the permissions are deleted, then accounts are disabled with **Update Account** and **Remove Child Data** operations.

For more details, see View Activity Log.

Assign Groups as Permissions

You can assign groups as permissions to a account using the **Request a new access** functionality of Oracle Access Governance. This allows you to request an access bundle containing permissions which equate to groups on the system.

When you request an access bundle, either directly or through an Oracle Access Governance role or policy, a provisioning operation, **Add Child Data**, is initiated which updates the groups in your instance with the permissions included in the referenced access bundle.

If you request groups for an identity that does not have corresponding account in the instance, then a new account is also created on the instance with **Create Account** operation.

For further details about permission assignment, refer to Request Access. To learn more about roles and policies, refer to Manage Roles, and Manage Policies.

Revoke Groups as Permissions

You can revoke group permissions from an account by removing the permission from the role, policy or access bundle to which it is assigned. In this case, the permission assignment is revoked from all users to whom the role, policy or access bundle is applied.

Another way to remove a permission would be by revoking role, policy or access bundle assignment from a specific account. This would be done using the revoke operation in access reviews.

If only permissions are different, then account remains enabled but **Add Child Data** and/or **Remove Child Data** operations are triggered to update the permissions for that account. If all the permissions are deleted, the accounts are disabled.

For further details about permission assignment, refer to Delete a Role, Delete a Policy, or Manage Access Bundles -> Delete an Access Bundle.

Example: Joiner Use Case for

Orchestrated System is used for managing accounts and groups across cloud service using Oracle Access Governance.

Scenario: A new employee joins as a *Sourcing Manager* in your team, and access to must be provisioned automatically with appropriate group membership. For this example, assume you have established an integration with Authoritative Source, Oracle HCM, and Oracle Access Governance data is synced with this new employee information. Use Oracle Access Governance to seamlessly manage accounts and group membership to .

1. Configure your instance with Oracle Access Governance using the steps defined in Configure Integration Between Oracle Access Governance and SAP Ariba.
2. Perform data load to reconcile existing accounts. **Full Data Load** for Day 0 and **Lookup Data Load** for Day N activities would trigger to ingest data from into Oracle Access Governance.
3. Configure your orchestrated system settings to further add matching rules, transformations, notification settings, and so on. For details, see Configure Settings for Orchestrated Systems.
4. In Oracle Access Governance **Access Controls** section, perform the following
   a. Create an Access Bundle for your Orchestrated System. Select appropriate *Groups*. For details, see Create Access Bundle.
   b. Create a policy within Oracle Access Governance and associate the access bundle with an identity collection, say *Sourcing_Managers*. Another way is to request access for this access bundle using the self-service functionality. For details, see Manage Policies and Request Access to a Resource.
5. If the access is requested, once approved, a new account is created with assigned group membership. **Create Account** and **Add Child Data** activities would be triggered to support account provisioning to your instance. If the provisioning operation is successful, then the new account is created within your instance.

Configure Integration Between Oracle Access Governance and

You can establish a connection between Oracle Access Governance and SaaS application as a Managed System. To configure, use **Orchestrated Systems** in the Oracle Access Governance Console.

Prerequisites

Before you install and configure the Orchestrated System, you should consider the following prerequisites and tasks.

Setup to Manage SAP Ariba Accounts - Configuring a New Integration Inbound Endpoint in SAP Ariba

Configure an inbound endpoint in SAP Ariba solution to enable provisioning and access management from Oracle Access Governance.

An end point consists of the URL and authentication information that controls access to the end point. Configure an inbound endpoint for initiating provisioning and access management operation from Oracle Access Governance.

Required Roles:

- Member of the Customer Administrator or Integration Admin group
- Group with the Administrator or Integration Admin role

Configure a New Integration Inbound Endpoint in SAP Ariba

1. Sign in to SAP Ariba instance with administrator credentials.
2. On the SAP Ariba Administrator dashboard, click **Manage**, and then **Administration**.
3. Expand the **Integration Manager** option and select **End Point Configuration**.
4. To create a new endpoint, click **Create New**.

   An **End Point Configuration - Create End Point** page opens.

5. In the **Name** field, enter a name for the end point.
6. Select the type as **Inbound**.
7. Navigate to the **HTTP Authentication section**, to use HTTP Basic Authentication.
   a. Enter the **user ID** in the **Sign In** field.
   b. Enter password in the **Password** field.

   You need to provide this information to configure orchestrated system in the Oracle Access Governance Console.
8. Click **Save**.

Setup for Data Load - Enable an API from the SAP Ariba API Developer Portal

You can reconcile data from your application to Oracle Access Governance by enabling the application API.

A single application cannot have access to more than one API. You may have only one application per realm/API combination.

Required Role: A user with the **Organization Admin** role requests approval for API access.

Create Application on the SAP Developer API Portal

You may skip the task if you already have an application.

1. Sign in to the developer portal.
2. Navigate to **Manage Applications**, then click the **+** plus sign.
3. Enter application name and description.
4. Click **Submit**.
   The application is created and is visible under the **Applications** list. A unique identifier for you application, **Application Key**, is also generated. You require this to configure with Oracle Access Governance

Request API Access for your application

1. Sign in to the developer portal as a user with the **Organization admin** role.
2. On the Administrator dashboard, click **Manage**, and then **Administration**.
3. Search the desired application in the application list that you want it enabled.
4. On the application setting page, click **Actions**, and then **Request API access**.
5. Complete the following application details:
   a. In the **API Names** drop-down list, choose the API name that you want to access.
   b. In the **Realm name**, choose the solution for which you want to enable the application.
   c. In the **AN-ID** field, enter your Ariba Network Identification Number (ANID).
   d. In the realm type, select **Production** or **Type**.
   e. Click **Submit**.

   The application is sent for approval. user with the **Organization Admin** role can approve the API access request for your application.

Generate OAuth Secret and Base64 Encoded Client ID and Secret for your Application

Once the application is approved by your administrator, you can generate OAuth Secret credentials to authenticate your API. Follow the given instructions:

1. Sign in to the developer portal as a user with the **Organization admin** role.
2. On the Administrator dashboard, click **Manage**, and then **Administration**.
3. Search the desired application in the application list that you want it enabled.
4. On the application setting page, click **Actions**, and then **Generate OAuth Secret**.
5. In the confirmation box, click **Submit**. The **OAuth Secret** and **Base64 Encoded Client and Secret** are displayed.
6. Copy the **OAuth Secret** and **Base64 Encoded Client and Secret** and save it to a secure location.

You'll need this to configure orchestrated system in the Oracle Access Governance Console.

Fetch OAuth Authentication Server URL

You must fetch OAuth Server URL for your API application to establish the connection.

You'll need the **Master Data Retrieval API for Sourcing** API Server URL to configure orchestrated system in the Oracle Access Governance Console.

Required Role: A user with the **Organization Admin** role in SAP Ariba Developer Portal.

1. Sign in to the developer portal as a user with the **Organization admin** role.
2. Go to **Discover** section, and navigate to **STRATEGIC SOURCING**.
3. Search for **Master Data Retrieval API for Sourcing**.
4. Copy the OAuth Server URL Prefix value, and append v2.
   Example: https://< OAuth Server URL >/v2

Fetch Partition ID and Variant values from WSDL

You need to fetch variant and partition ID of the realm to maintain parameter for Master Data Set.

1. Sign in to the Ariba application with admin credentials.
2. In the Ariba Administrator dashboard, click **Manage** and select **Administration** from drop-down.
3. Expand **Integration Manager** and select **Integration Configuration**.
4. Search for **Import Users** web service, Task name=`Import User`.
5. Click and open the **Import Users** web service.
6. Click **View WSDL** and search for vrealm in WSDL.
7. If you find **vrealm_1234** in WSDL, you can use *vrealm_1234* as Variant and *prealm_1234* as Partition.

Configure

You can establish a connection between and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Identity orchestrations are set up from the Oracle Access Governance Console. Go to the Orchestrated Systems page to integrate with Oracle Access Governance.

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select System

On the **Select system** step of the workflow, you can specify the type of system that you want to integrate.

You can search for the required system by name using the **Search** field.

1. Select .
2. Click **Next**.

Enter Details

In the *Enter Details* step, give a meaningful name to your orchestrated system, add a supporting description, and determine if you can use this system as an authoritative source or for managing permissions. For , you can use Oracle Access Governance to manage permissions for identity accounts.

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.

> **Note:**
>
> A message is displayed on the page indicating that Oracle Access Governance can manage permissions for this system, enabling provisioning of accounts.

3. Click **Next**.

Add Owners

In this step, add primary and additional owners for your orchestrated system.

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account Settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager

For this orchestrated system, the identity accounts can only be disabled and not be deleted. So, the choices to select for mover and leaver case will be grayed out.

Integration Settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to .

Fill the configuration information as explained in the following table, and then click **Add**.

Table 5-40 Integration Details

Field	Description	Example	Reference
What is the realm name hosting SAP Ariba?	Enter the unique realm name for your solution. Typically, you can find this in the URL in the format `s1.ariba.com/sourcing/Main/xxxxxrealm=MyRealm-T`	`MyRealm-T`	How do I find out my realm name?
What is the API Key to be used for loading data?	Enter the unique API application key for the application you created	`123abc12345ABXX`	• API's Application Key • Finding Your Application's Application Key and OAuth Client ID
What is Oauth Client ID?	Enter client ID for your API Application. This information is visible when you generate OAuth credentials for your application.	`123ABC12345acxx`	Generate OAuth Credentials

Table 5-40 (Cont.) Integration Details

Field	Description	Example	Reference
What is OAuth Client Secret?	Enter client Secret for your API Application. This information is visible when you generate OAuth credentials for your application.	`123ABC12345aTT`	Generate OAuth Credentials
What is the authentication server URL to validate the client?	Enter the OAuth Server URL and append **v2**	`https://<OAuth Server URL>/v2`	Fetch OAuth Authentication Server URL
What is username?	Enter the user id for HTTP Basic authentication	`ag24sapariba`	Setup to Manage SAP Ariba Accounts - Configuring a New Integration Inbound Endpoint in SAP Ariba
What is password?	Enter the password for HTTP Basic authentication	`ag24sapariba`	Setup to Manage SAP Ariba Accounts - Configuring a New Integration Inbound Endpoint in SAP Ariba
Confirm password	Re-enter the password for confirmation.	`ag24sapariba`	Setup to Manage SAP Ariba Accounts - Configuring a New Integration Inbound Endpoint in SAP Ariba
What is the tenancy's unique partition name?	Enter the unique partition name for your realm	`prealm_1234`	Fetch Partition ID and Variant values from WSDL
What is the tenancy's unique variant name?	Enter the unique variant name for your realm.	`vrealm_1234`	Fetch Partition ID and Variant values from WSDL

Finish Up

Review and configure your configuration setup. You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load.

Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with the system.

Integration Reference

Lists certified components, supported operations, configuration modes, default out-of-the-box attributes for the integration between and Oracle Access Governance.

Components Certified for Integration with Oracle Access Governance

Lists the software components and their versions required for integrating with Oracle Access Governance.

Certified Components

Table 5-41 Certified Components

Component Type	Component
Managed System	SAP Ariba
API	• **For Account Provisioning**: Import Users web service, `version 1` • **For Reconciliation:** Master Data Retrieval API for Sourcing, `version 1.0.0`

Supported Configuration Modes for

You can use Oracle Access Governance integrations to set up different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Supported Modes

The Orchestrated System supports the following modes:

- **Managed System**
  You can manage user accounts and groups from Oracle Access Governance.

Supported Operations when Provisioning to

Orchestrated System supports the user management and group management operations.

The Orchestrated System supports the following account operations when provisioning a user:

- Create User
- Update User
- Enable User
- Disable User
- Assign Groups
- Revoke Groups

Default Supported Attributes

Oracle Access Governance supports the following default attributes.

These attributes are mapped depending on the direction of the connection, for example:

- Data being ingested by Oracle Access Governance from :

`UniqueName` will map to `User login`

- Data for the user accounts as identities being ingested by Oracle Access Governance from .

  `PasswordAdapter` will map to `User Type`

Table 5-42 Default Attributes for

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Transformation Supported?	Mandatory?
User	UniqueName	uid	Unique Id		
	UniqueName	name	User login	Yes	Mandatory
	Name_en	displayName	Name	Yes	Mandatory
	EmailAddress	email	Email		Mandatory
	PasswordAdapter	userType	User type	Yes, with 'Enterprise User'	Mandatory
	Organization	organization	Organization	Type: ADMIN	
	Supervisor	supervisor	Supervisor	No	
		supervisorUserType	Supervisor user type	No	
	LocaleID	locale	Locale	No	
	Phone	phone	Phone number	No	
	active	status	Status	No	
	TimeZoneID	timezone	Timezone	No	
Groups		groups	Groups	No	

SAP S4HANA

Integrate Oracle Access Governance with

Oracle Access Governance enables **API-based** seamless integration with for enabling identity orchestration, automating onboarding of accounts and roles, reconciliation of accounts. Oracle Access Governance supports account management and role management for accounts as a **Managed System**.

is a comprehensive cloud-based procurement and spend management service that helps businesses streamline and optimize their procurement processes, from sourcing to payment. With this integration, you can update, enable, and disable identity accounts. You can assign or revoke roles for accounts from Oracle Access Governance.

Overview: Orchestrated System

You can establish a connection between and Oracle Access Governance by entering connection details and configuring the orchestrated system. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

Integration Architecture Overview

You can perform full data load for accounts in . Once a connection is established, you can perform remediation tasks for user accounts and roles.

integration leverages the following SOAP APIs for full incremental data load.

- Use the **Business Users Read** SOAP API to read data, and perform full data load in Oracle Access Governance.
- Use the **Business Users** SOAP API to update account attributes and assign roles to accounts from Oracle Access Governance.

Functional Overview: Use Cases Supported for Integration

integration supports account management and role management for accounts. The orchestrated system supports management of accounts for *Business Users* .

- Configure Orchestrated System

  See Configure Integration Between Oracle Access Governance and SAP S4Hana
- Match Identity and Account Attributes using Correlation Rules

  Review or Configure the matching rules to match the identity and account data and build a composite identity profile. To view the default matching rule for this orchestrated system, see Default Supported Attributes.
- Load Data

  Ingest accounts and roles that can be managed by Oracle Access Governance
- Update Account

  Modify account attributes, such as locking or unlocking the account by editing the **LockedIndicator** attribute account.

  > **Note:**
  >
  > As a user with `AG_ServiceDesk_Admin` role, use the **Edit Account** feature from the **Manage Identities** page to update the account.
- Enable Account

  If only permissions are different, then account remains enabled but **Add account or permission data** and/or **Remove account or permission data** operations are triggered in the Orchestrated System to update the permissions for that account.
- Disable Account

  If all the permissions are deleted, then accounts are disabled with **Update Account** and **Remove account or permission data** operations.
- Assign Roles as Permissions
- Revoke Roles as Permissions

Example: Use Case for

Orchestrated System is used for managing accounts and roles across cloud service using Oracle Access Governance.

Scenario: A policy violation is detected due to multiple failed sign-in attempts for an account. To take immediate action, a user with the `AG_ServiceDesk_Admin` role can modify account attributes immediately without undergoing business approvals. Use Oracle Access Governance to seamlessly manage accounts and role assignments to . In this scenario, you would lock an account of the user.

1. Configure your instance with Oracle Access Governance using the steps defined in Configure Integration Between Oracle Access Governance and SAP S4Hana.
2. Perform data load to reconcile existing accounts. **Full Data Load** for Day 0 and **Lookup Data Load** for Day N activities would trigger to ingest data from into Oracle Access Governance.
3. Configure your orchestrated system settings to further add matching rules, transformations, notification settings, and so on. For details, see Configure Settings for Orchestrated Systems.
4. As a `AG_ServiceDesk_Admin` user, from the **Manage Identities** page, perform the following

   a. From the **Identities** list, select the **Actions** icon and select **View details**. The Identity details page is displayed with the **Permissions** tab selected by default.
   b. Select the **Accounts** tab.
   c. Select the **Actions** icon corresponding to the account that you want to edit.
   d. Select **Edit Account**.
   e. Clear the **Account locked** check box and save the details.
5. This would trigger **Update Account** and **Remove account or permission data** on the orchestrated system. If the provisioning operation is successful, then the user account is locked for your instance.

Configure Integration Between Oracle Access Governance and

You can establish a connection between Oracle Access Governance and SaaS application as a Managed System. To configure, use **Orchestrated Systems** in the Oracle Access Governance Console.

Prerequisites

Before you install and configure the Orchestrated System, you should consider the following prerequisites and tasks.

Setup to Authenticate for Data Exchange - Create a Communication User

Create a communication user to authenticate for reconciliation and provisioning in Oracle Access Governance. Use the same communication user to perform integration operations in Oracle Access Governance.

You need the **Administrator** role to use the Communication Management applications.

Create a Communication User in

1. Sign in to **Cloud** application with administrator credentials.
2. From the **Communication Management** catalog, choose the **Communication Systems** application.
3. Click **New** to create a new communication user.

   The *Create Communication User* page is displayed.
4. Enter User Name, Description, and Password.
5. Click **Propose Password** to get a system-generated password. Save the credentials for authentications for your communication user.
6. Click **Create**.

Create Communication System

Perform the following steps to create a communication system and assign a communication user to the communication system

1. Sign in to Cloud application with administrator credentials.
2. From the **Communication Management** catalog, choose the **Communication Systems**.
3. Click **New** to create a new communication system.
4. Enter System ID and System Name, and then click **Create**.
5. Under **Technical Data**, enter the Host Name of your SAP S/4HANA Cloud tenant in the following format: `<tenant ID>.s4hana.ondemand.com`
6. Click **User for Inbound Communication** tab, then click the add (+) icon.
7. Assign the communication user you created, and select the authentication method as **User ID** and **Password**.
8. Click **Save**.

Create a Communication Arrangement

Create a communication arrangement by setting up endpoint URLs to call the SOAP service.

Inbound service allows you to configure settings for sending data from Oracle Access Governance to .

Create a New Communication Arrangement

1. Log in to Cloud application with administrator credentials.
2. From the **Communication Management** catalog, choose the **Communication Arrangements**.
3. Click **New** to create a new communication arrangement.
4. Select communication scenarios, as follows

- **SAP_COM_0093** Identity Management Integration
- **SAP_COM_0193** Identity Provisioning Integration

5. Enter an arrangement name, and click **Create**.
6. Select **Communication Arrangement** in the list. The inbound communication user is automatically assigned.
7. Under **Inbound Services**, the endpoint URLs to call the SOAP service in the following format

Option	Description
Business User - Read	`https://<S4HANA tenant ID>-api.s4hana.ondemand.com/sap/bc/srt/scs_ext/sap/querybusinessuserin` For more details, see Business User - Read
Business User - Update	`https://<S4HANA tenant ID>-api.s4hana.ondemand.com/sap/bc/srt/scs_ext/sap/managebusinessuserin` For more details, see Business User

8. Click **Save**.

   WSDLs can be downloaded from this arrangement once saved.
9. Expand the **Integration Manager** option and select **End Point Configuration**.
10. To create a new endpoint, click **Create New**.

    An **End Point Configuration - Create End Point** page opens.
11. In the **Name** field, enter a name for the end point.
12. Select the type as **Inbound**.
13. Navigate to the **HTTP Authentication section**, to use HTTP Basic Authentication.
    a. Enter the **user ID** in the **Sign In** field.
    b. Enter password in the **Password** field.

    You need to provide this information to configure orchestrated system in the Oracle Access Governance Console.
14. Click **Save**.

Configure

You can establish a connection between and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Identity orchestrations are set up from the Oracle Access Governance Console. Go to the Orchestrated Systems page to integrate with Oracle Access Governance.

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select System

On the **Select system** step of the workflow, you can specify the type of system that you want to integrate.

You can search for the required system by name using the **Search** field.

1. Select .
2. Click **Next**.

Enter Details

In the *Enter Details* step, give a meaningful name to your orchestrated system, add a supporting description, and determine if you can use this system as an authoritative source or for managing permissions. For , you can use Oracle Access Governance to manage permissions for identity accounts.

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **Name** field.
2. Enter a description for the system in the **Description** field.

> **Note:**
>
> A message is displayed on the page indicating that Oracle Access Governance can manage permissions for this system, enabling provisioning of accounts.

3. Click **Next**.

Add Owners

In this step, add primary and additional owners for your orchestrated system.

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account Settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager

For this orchestrated system, the identity accounts can only be disabled and not be deleted. So, the choices to select for mover and leaver case will be grayed out.

Integration Settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to .

Fill the configuration information as explained in the following table, and then click **Add**.

Table 5-43 Integration Details

Field	Description	Example	Reference
What is the base URL to connect to?	Enter the base URL for your SAP application. Your base URL is in the format `https://<hostname.s4hana.cloud.sap>:<port>`	`https://my-sap-system.s4hana.cloud.sap`	
What is the username to use for loading data?	Enter the communication username that you created in the prerequisite.	`John_Rim`	Setup to Authenticate for Data Exchange - Create a Communication User
What is the password?	Enter password for your communication system user for authentication.	`123ABC12345@cxx`	Setup to Authenticate for Data Exchange - Create a Communication User
Confirm password	Confirm the password.	`123ABC12345@cxx`	

Finish Up

Review and configure your configuration setup. You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load.

Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with the system.

Integration Reference

Lists certified components, supported operations, configuration modes, default out-of-the-box attributes for the integration between and Oracle Access Governance.

Components Certified for Integration with Oracle Access Governance

Lists the software components and their versions required for integrating with Oracle Access Governance.

Certified Components

Table 5-44 Certified Components

Component Type	Component
Managed System	
API	• **For Account Provisioning**: `ManageBusinessUserIn, v1.0.0` • **For Reconciliation:** `QueryBusinessUserIn, v1.0.0`

Supported Operations when Provisioning to

Orchestrated System supports the user management and role management operations. The orchestrated system supports management for Business User and Workforce User.

The Orchestrated System supports the following account operations when provisioning a user:

- Update Account (only **LockedIndicator**)
- Enable Account
- Disable Account
- Add Role
- Remove Role

For more details, see Oracle Access Governance Integration Functional Overview and Integrate Oracle Access Governance with SAP S4Hana.

Supported Configuration Modes for

You can use Oracle Access Governance integrations to set up different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Supported Modes

The Orchestrated System supports the following modes:

- **Managed System**
  You can manage user workforce and business user accounts and roles from Oracle Access Governance.

Default Supported Attributes

Oracle Access Governance supports the following default attributes.

These attributes are mapped depending on the direction of the connection, for example:

- Data being ingested by Oracle Access Governance from :

  `User login` will map to `Employee user name`

Table 5-45 Default Attributes for

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name
User	PersonID	uid	Unique Id
	UserLogin	name	User name
	UserID	userID	User ID
	PersonExternalID	personExternalID	person external id
	FirstName	firstName	First name
	LastName	lastName	Last name
	PersonFullName	displayName	Name
	MiddleName	middleName	Middle name
	EmailAddress	email	Email
	PersonUUID	personUUID	Person UUID
	StartDate	startDate	Start date
	EndDate	endDate	End date
	LockedIndicator	accountisLockedout	Account locked
	status	status	Status
Roles	_ROLE_	roles	Roles

SAP SuccessFactors

Overview: Integrate Oracle Access Governance with

Oracle Access Governance can be integrated with , enabling identity orchestration, including on-boarding of identity (user) data, and provisioning of accounts.

can be integrated with Oracle Access Governance as an authoritative source or managed system, allowing you to reconcile human capital management (HCM) details, and provision and manage identities and accounts.

Integration Architecture Overview

You can perform full data load for accounts in . Once a connection is established, you can perform remediation and management tasks for user accounts, and static groups.

Oracle Access Governance uses OAuth to authorize access to the OData API, This enables Oracle Access Governance to perform reconciliation and provisioning tasks on HCM details held in .

Functional Overview: Use Cases Supported for Integration

integration supports management of accounts by Oracle Access Governance, including the following use cases.

- Configure Orchestrated System

  See Configure Integration Between Oracle Access Governance and SAP SuccessFactors.

- Match Identity and Account Attributes using Correlation Rules

  Review or configure matching rules to match the identity and account data and build a composite identity profile. To view the default matching rule for this orchestrated system, see Default Matching Rules.

- Load Data

  Ingest accounts and groups that can be managed by Oracle Access Governance.

- Create Account

  Ingest account data from your orchestrated system or request an access for an identity. This allows you to provision entitlements (Employee).

- Update Account

  Update account details by assigning or removing permissions.This allows you to update entitlements (Group).

- Enable/disable Account

  Enable or disable an account (Users) associated with an identity. This will either remove or restore accesses for the account.

Prerequisites

Before you install and configure a Orchestrated System, you should consider the following prerequisites and tasks.

1. Your system is certified with Oracle Access Governance. Refer to Components Certified for Integration with Oracle Access Governance for details of the versions supported.

Configure

You can establish a connection between and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

The Orchestrated Systems page of the Oracle Access Governance Console is where you start configuration of your orchestrated system.

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.

2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to integrate with Oracle Access Governance.

You can search for the required system by name using the **Search** field.

1. Select .
2. Click **Next**.

Add details

Add details such as name, description, and configuration mode.

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owners

Add primary and additional owners to your orchestrated system to allow them to manage resources.

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

Outline details of how to manage account settings when setting up your orchestrated system including notification settings, and default actions when an identity moves or leaves your organization.

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

Enter details of the connection to your system.

1. On the **Integration settings** step of the workflow, enter the details required to allow Oracle Access Governance to connect to your system.

Table 5-46 Integration settings

Parameter Name	Mandatory?	Description
Account Type	Yes	Select what type of account you want to integrate from your system. This can be one of the following options: • **User** • **Employee** This option applies when your orchestrated system is running in authoritative source mode. The identity attributes supported will vary dependent on account type. See Supported Attributes for details of available identity attributes in each case.
What is the base URL for SuccessFactors?	Yes	The URL to access your system. For example: `<http|https://<hostname>/successfactors.com:<port>`
What is the company ID?	Yes	Company ID is a short string of characters that identifies each system. It is used during login to validate your access token and for provisioning operations.
What is the authentication server url to validate the client?	Yes	URL of the authentication server that validates credentials for the target system. For example: `https://<hostname>/successfactors.com/oauth/token`).
What is the authorization server url?	Yes	Required only when a private key is used to generate a SAML token, for example: `https://<hostname>/successfactors.com/oauth/idp`
What is the client ID?	Yes	ID issued by the authorization server when registering your client.
What is the authentication mechanism?	Yes	Select from one of the following: • SAML Assertion Token • Private Key
What is the SAML assertion?	No	If authentication mechanism selected is **SAML Assertion Token** then enter the SAML token issued by the trusted IDP for your client.
What is the private key?	No	If authentication mechanism selected is **Private Key** then enter the encrypted private key .
What is the username?	No	Required only when a private key is used to generate a SAML token

2. Click **Add** to create the orchestrated system.

Finish Up

Finish up configuration of your orchestrated system by providing details of whether to perform further customization, or activate and run a data load.

The final step of the workflow is **Finish Up**.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with the system.

Components Certified for Integration with Oracle Access Governance

The components that you can integrate with are listed below.

Table 5-47 Certified Components

Component Type	Component
System	

Supported Configuration Modes for Integrations

Oracle Access Governance integrations can be setup in different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Orchestrated System supports the following modes:

- **Authoritative Source**
  You can use as an authoritative (trusted) source of identity information for Oracle Access Governance.
- **Managed System**
  You can manage accounts, and groups.

Supported Operations When Provisioning To

When you provision an account from Oracle Access Governance to certain operations are supported.

The Orchestrated System supports the following account operations when provisioning a user:

- Create User
  Only an Employee with an Employee ID is supported.
- Update User
- Enable User
  Employee cannot be enabled. The underlying User associated with the Employee will be enabled.

- Disable User
  Employee cannot be disabled. The underlying User associated with the Employee will be disabled.
- Add Group

> **Note:**
>
> Currently only **static groups** are supported. Dynamic groups are not supported.

- Remove Group

> **Note:**
>
> Currently only **static groups** are supported. Dynamic groups are not supported.

For more details see Oracle Access Governance Integration Functional Overview and Integrate with SAP SuccessFactors.

Default Supported Attributes

Oracle Access Governance supports the following default attributes.

Table 5-48 Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Supported Identity attribute displayname	Module	User/
User	UserEntity.userId	uid	User & Employee		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported	Module	User/
			Employee Oracle Access Governance Identity attribute displayname		
			queld		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee
			Oracle Access Governance Identity attribute display name	
	UserEntity.username	name	User & Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee
		oracleAccessGovernanceIdentityattributedisplayname		
		username		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee
	UserEntity.email	email	User & Employee	Email

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee Access Governance Identity attribute displayname
	UserEntity.country	country	User & Employee	Country

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee Access Governance Identity attribute displayname
	UserEntity.state	state	User & Employee	State

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module User/Employee Role Access Governance Identity attribute display name
	UserEntity.city	city	User & Employee City

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee Access Governance Identity attribute display name
	UserEntity.citizenship	citizenship	User & Employee	Citizens

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported	Module	User/ Employee
			OracleAccessGovernanceIdentityattributedisplayname		
			hip		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee
	UserEntity.empId	employeeNumber	User & Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute displayname	Supported Employee Module	User/
			number		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module User/Employee Oracle Access Governance Identity attribute display name
	UserEntity.jobLevel	jobLevel	JUser & Employee jobLevel

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Corresponding Supported Module User/Employee Attribute Display Name
	UserEntity.married	married	married User & Employee

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee Access Governance Identity attribute displayname
	UserEntity.status	status	User & Employee	Status

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee
	UserEntity.businessPhone	phone	User & Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee	Access Governance Identity attribute display name
	UserEntity.cellPhone	mobile	User & Employee	Mobile	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee Oracle Access Governance identity attribute displayname
	UserEntity.firstName	firstName	User & Employee	First name

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee
		oracleAccessGovernanceIdentityattributedisplayname		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee Correlation Attribute	Display name
	UserEntity.lastName	lastName	User & Employee	L	Last name

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee	Oracle Access Governance Identity attribute display name
	UserEntity.gender	gender	User & Employee		Gender

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee
	UserEntity.salutation	salutation	User & Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Supported	Employee	Module	User/

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/ Employee	Oracle Access Governance Identity attribute display name
	UserEntity.middleName	middleName	User & Employee	Middle name	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee
			Oracle Access Governance Identity attribute display name	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee	Oracle Access Governance Account Attribute Display name
	PerPersonal.personIdExternal	externalPersonId	User & Employee	External	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Owner Employee	Module	User/
			accessGovernanceIdentityattributedisplayname		
			personid		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Cr	Supported	Employee	Module	User/
			a c l e A c c e s s G o v e r n a n c e I d e n t i t y a t t r i b u t e d i s p l a y n a m e				
Employee	EmpJob.managerId	managerUid	M a n a g e r		Employee		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute displayname	Supported Module	User/ Employee
	EmpJob.company	company	Œcompany		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Supported Identity attribute display name	Module	User/ Employee
	EmpJob.department	department	Department		Employee

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Corr	Supported	Employee	Module	User/
			aclel				
			eAcc				
			essG				
			overn				
			ancel				
			dent				
			ityat				
			trib				
			uted				
			displ				
			ayna				
			me				
			nt				

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute displayname	Supported Module	Employee User/
	EmpJob.jobCode	jobCode	JobCode	Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Oracle Access Governance Identity attribute display name	Module	User/ Employee
	EmpJob.division	division	Division	Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute displayname	Supported Module	User/ Employee
	EmpJob.location	location	Location		Employee

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Cancel Identity attribute displayname	Supported Module	Employee User/
	EmpJob.businessUnit	businessUnit	Business	Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported	Module	User/
			Employee Orac leAccessGovernanceIdentityattributedisplayname		
			unit		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee Access Governance Identity attribute displayname
	EmpJob.payScaleArea	payScaleArea	PeopleSoft	Employee payscale

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Employee	Module	User/
			Oracle Access Governance Identity attribute display name		
			area		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Supported Employee Module User/ Role Access Governance Identity attribute displayname
	EmpJob.employmentType	employmentType	Employee

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Employee Module User/ Oracle Access Governance Identity attribute display name
			ntty pe

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute displayname	Supported Employee	Module	User/
	EmpJob.isFulltimeEmployee	isFulltimeEmployee	FEmployee ull-tim			

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Employee Module User/
			oracleAccessGovernanceIdentityattributedisplayname
			eemploy

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Corresponding Supported Employee Module User/
			accessGovernanceIdentityattributedisplayname
			ee

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Supported Account Attribute	Module	User/ Employee
	EmpEmployment.startDate	startDate	StartDate	Employee	

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Corresponding Supported Module User/Employee Access Governance identity attribute displayname
			te

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Supported Employee Module User/ Account Identity attribute displayname
	EmpEmployment.endDate	endDate	Employee enddate

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Supported Employee	Module	User/
	EmpEmployment.isContingentWorker	isContingentWorker	Continge	Employee		

Table 5-48 (Cont.) Default Attributes for - Authoritative Source

Entity	Account Attribute	Oracle Access Governance Account Attribute	Supported Module	User/Employee
		oracleAccessGovernanceIdentityattributedisplayname		
		network		

Table 5-49 Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Orchestrated System Updateable Field
User	UserEntity.userId	uid	UNo no iquel d

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field	Display name
	UserEntity.username	name	User Login	No	

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Transformation	Updateable Field
			in		

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field
	UserEntity.password	password	No	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	OTUpdateable Field
	UserEntity.email	email	EYYes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field	Country
	UserEntity.country	country	Yes	No	

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Field	Updateable Field
	UserEntity.state	state	No	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Transformation	Updateable Field
	UserEntity.city	city	City	No	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field
	UserEntity.citizenship	citizenship	Yes	No

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Transformation	Updateable Field
			hip		

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute Flag	Updateable Field
	UserEntity.empId	employeeNumber	No	No

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	OTUpdateable Field
			rr aa cn l s ef Ao cr cm ea st si Go on v e r n a n c e l d e n t i t y a t t r i b u t e d i s p l a y n a m e
			n u m b e r

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Orchestrated System Account Attribute Updateable Field
	UserEntity.jobLevel	jobLevel	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable	Field Name
	UserEntity.married	married	No	Yes	Marital Status

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	OTUpdateable Field
			rraacnlsefAocrcmeastsiGoonvernanceldentityattributedisplayname
			tatus

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Field / Updateable Field / Display name
	UserEntity.status	status	Yes / Yes / status

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Field	Updateable Field
	UserEntity.firstName	firstName	Yes	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Orchestrated System Updateable Field
			Display name

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field	Display name
	UserEntity.lastName	lastName		Yes	Last name

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Orchestrated System Account Attribute Updateable Field
	UserEntity.gender	gender	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Transformation	Updateable Field
	UserEntity.salutation	salutation	Salutati	No	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Definition Display Name	Updateable Field
				on

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Definition Attribute Display Name	Updateable Field
	UserEntity.middleName	middleName	Middle Name	No

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field	Display name
	UserEntity.businessPhone	phone	No	Yes	phone

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field
	UserEntity.cellPhone	mobile	No	Yes

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute	Updateable Field	Display name
	UserEntity.fax	fax	No	Yes	

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Correlation Attribute Flag	Updateable Field	Display name
	PerPersonal.personIdExternal	externalPersonId	No	External	

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Updateable Field
			Orchestrated Account Display Name
			person id

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Orchestrated System Updateable Field
Group	groupId_staticGroup	groups	No groups

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	OTUpdateable Field
Dynamic Group	groupId_dynamicGroup	dynamicGroups	DNNo yo n a m i c G

Table 5-49 (Cont.) Default Attributes for - Managed System

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name	Transformation	Updateable Field
			roups		

> **Note:**
>
> For Managed System mode, only User related attributes are supported. Employee provisioning is not supported.

Default Matching Rules

In order to map accounts to identities in Oracle Access Governance you need to have a matching rule for each orchestrated system.

The default matching rule for the orchestrated system is:

Table 5-50 Default Matching Rules

Mode	Default Matching Rule

Authoritative Source
Identity matching checks if incoming identities match an existing identity or are new.
	Screen value:
	`User login = Employee user name`
	Attribute name:
	`Account.UserEntity.username = Identity.Name`

Managed System
Account matching checks if incoming accounts match with existing identities.
	Screen value:
	`User login = Employee user name`
	Attribute name:
	`Account.UserEntity.username = Identity.Name`

The Database User Management (DB2) connector integrates Oracle Access Governance with IBM DB2 database user management tables. You can establish a connection between IBM DB2 and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Prerequisites

Before you install and configure a Database User Management (DB2) orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

The DB2 system can be any of the following:

- **IBM DB2 UDB 9.x**.

Supported Operations

The Database User Management (DB2) orchestrated system supports the following operations:

- Create user
- Add schema

- Revoke schema
- Add tablespace
- Revoke tablespace

Create a DB2 System User Account for Database User Management (DB2) Orchestrated System Operations

Oracle Access Governance requires a user account to access the DB2 system during service operations. Depending on the system you are using, you can create the user in your system and assign specific permissions and roles to the user.

For DB2 database:

1. Create a DB2 user `agserviceuser` at the OS level.
2. Assign the following permissions and roles to the service user created:

```
GRANT SELECT on TABLE syscat.schemata TO agserviceuser;

GRANT SELECT on TABLE syscat.tablespaces TO agserviceuser;

GRANT CREATEIN,DROPIN,ALTERIN ON SCHEMA 'SCHEMA_NAME' TO agserviceuser;

GRANT
CONNECT,BINDADD,DBADM,CREATETAB,CREATE_NOT_FENCED_ROUTINE,IMPLICIT_SCHEMA,LOAD,CREATE
_EXTERNAL_ROUTINE,QUIESCE_CONNECT ON DATABASE TO agserviceuser;
```

Configure

You can establish a connection between IBM DB2 and Oracle Access Governance by entering connection details and configuring your database environment. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon, select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of database you would like to onboard.

1. Select **Database User Management (DB2)**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.

3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable

- Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the IBM DB2 database.

1. In the **Easy Connect URL for Database** field, enter the connect string for the database you want to integrate with Oracle Access Governance, in the format *host/port/database service/sid*.
2. In the **User Name** field, enter the IBM DB2 user you will use to connect to the database. This is the user you created in Create a DB2 System User Account for Database User Management (DB2) Orchestrated System Operations.
3. Enter the password of the IBM DB2 database user in the **Password** field. Confirm the password in the **Confirm password** field.
4. In **Connection Properties** enter any connection properties in the format *prop1=val1#prop2=val2*
5. In **Custom Jar Details** enter the jar name and the jar checksum in the format *<jarName>::<jarChecksum*.
   For example:

   ```
   db2jcc.jar::c8520f145b428b1133b771bb2c70a6f0f546c9f0655f9de5de2e7b64d5ede786911ad50b5
   43846154fe373dead78d38fb6dded560e0de4c4e8ccbbf0a06b6c1e
   ```

   For more information on custom jar support, refer Custom Jar Support.
6. Check the right hand pane to view **What I've selected.** If you are happy with the details entered, click **Add**.

Finish up

On the **Finish Up** step of the workflow, you are asked to download the agent you will use to interface between Oracle Access Governance and IBM DB2 database. Select the **Download** link to download the agent zip file to the environment in which the agent will run.

After downloading the agent, follow the instructions explained in the Agent Administration article.

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

For DB2 Database system to integrate with Oracle Access Governance, driver jar needs to be registered with the agent. For more information, refer Custom Jar Support.

Prerequisites

Before you install and configure a Database User Management (Microsoft SQL Server) orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

The Microsoft SQL Server system can be any one of the following:

- **Microsoft SQL Server 2005, 2008, 2012, 2014, 2016, 2019**

Supported Modes

Database User Management (Microsoft SQL Server) orchestrated system supports **Managed System** mode.

Supported System Operations

The Database User Management (Microsoft SQL Server) orchestrated system supports the following Microsoft SQL Server operations:

- Create UserName
- Create UserLogin
- Change UserLogin password
- Delete UserLogin
- Assign Roles to a userName
- Revoke Roles from a userName

Create a System User Account for Database User Management (MSSQL) System Operations

1. Verify the following requirements for your Microsoft SQL Server installation before configuring the orchestrated system.
   - The Microsoft SQL Server TCP/IP port is enabled. The default port is 1433. To enable the TCP/IP port:
     a. Open the Microsoft SQL Server Configuration Manager.
     b. Click **SQL Server Network Configuration**.

c. Click **Protocols for MSSQLSERVER**.

d. In the right frame, right-click **TCP/IP** and then click **Enable**.

- Mixed mode authentication is enabled.
- The TCP/IP port is not blocked by a firewall.

2. Create Login using the following query:

```
Create LOGIN serviceuser with PASSWORD='password' , DEFAULT_DATABASE
=DBname
GO
```

3. Create a user using the following query:

```
USE DBname;
Create USER serviceuser with LOGIN serviceuser;
GO
```

4. Assign the following permissions and roles to the created user:

```
ALTER ROLE db_datawriter ADD MEMBER serviceuser;
ALTER ROLE db_datareader ADD MEMBER serviceuser;
ALTER ROLE db_accessadmin ADD MEMBER serviceuser;
ALTER ROLE db_owner ADD MEMBER serviceuser;
exec sp_addsrvrolemember 'serviceuser', 'securityadmin';
```

Configure

You can establish a connection between Microsoft SQL Server and Oracle Access Governance by entering connection details. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard.

1. Select **Database User Management (MSSQL)**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.

2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owner

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the Microsoft SQL Server.

1. In the **Easy connect URL for the database** field, enter an easy connect URL to connect to the Microsoft SQL Server database, using the following syntax `jdbc:sqlserver://[host]:[port];[databaseName];[encrypt];[trustServerCertificate]`. For further information, consult the JDBC driver documentation..
2. In the **Username** field, enter the administration username you will use to connect to the Microsoft SQL Server database.
3. Enter the password for the adminstration user in the **Password/Confirm password** fields.
4. In the **Connection properties** field, enter any connection properties that will be used to configure a secure connection. They should be key value pairs in the following format: `key1=val1#key2=val2`.
5. The agent for this orchestrated system requires a Microsoft SQL Server driver jar in the classpath. Details of how this is used by the agent can be found in Custom Jar Support. The Microsoft SQL Server jar name and checksum should be in the format of `<jarName>::<jarChecksum>`.
6. Update **Database inclusion list** with one or more database names that data should be included in the dataload.
7. Update **Database exclusion list** with one or more database names that data should be excluded from the dataload.
8. Click **Add** to create the orchestrated system.

Finish up

On the **Finish Up** step of the workflow, you are asked to download the agent you will use to interface between Oracle Access Governance and Microsoft SQL Server database. Select the **Download** link to download the agent zip file to the environment in which the agent will run.

After downloading the agent, follow the instructions explained in the Agent Administration article.

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with a Microsoft SQL Server system.

The Database User Management (MySQL) connector integrates Oracle Access Governance with database user management tables in Oracle Database. You can establish a connection between Oracle Database (MySQL) and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

Prerequisites

Before you install and configure a Database User Management (MySQL) orchestrated system, you should consider the following prerequisites and tasks.

Certified Components

The MySQL system can be any one of the following:

- **MySQL 5.x**.

Supported Operations

The Database User Management (MySQL) orchestrated system supports the following operations:

- Create user
- Reset password
- Add privileges
- Revoke privileges

Create a User Account for Database User Management (MySQL) Orchestrated System Operations

Oracle Access Governance requires a user account to access the MySQL system during service operations. Depending on the system you are using, you can create the user in your system and assign specific permissions and roles to the user.

For MySQL:

1. Create a user `agserviceuser` user using the following query:

```
CREATE USER agserviceuser IDENTIFIED BY 'password';
```

2. Assign the following permissions and roles to the created user using the following query:

```
GRANT, SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER ON *.* TO 'agserviceuser';
```

Configure

You can establish a connection between Oracle Database User Management (MySQL) and Oracle Access Governance by entering connection details and configuring your database environment. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select System

On the **Select system** step of the workflow, you can specify which type of database you would like to onboard.

1. Select **Database User Management (MySQL)**.

Enter Details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the MySQL database.

1. In the **Easy Connect URL for Database** field, enter the connect string for the database you want to integrate with Oracle Access Governance, in the format *host/port/database service/sid*.
2. In the **User Name** field, enter the dastabase user you will use to connect to the database. This is the user you created in Create a User Account for Database User Management (MySQL) Orchestrated System Operations.
3. Enter the password of the target database user in the **Password** field. Confirm the password in the **Confirm password** field.
4. In **Connection Properties** enter any connection properties in the format *prop1=val1#prop2=val2*
5. In **Custom Jar Details** enter the jar name and the jar checksum in the format *<jarName>::<jarChecksum>*.
   For example:

   ```
   mysql-connector-
   j-8.0.32.jar::ca7894157bc91a5a9f46eac954795450a9565c7693391dc25c2ec7ac6c86a43e695e9a2
   a6a141c21c700611701543395b52ffb3b4f6b2dab613d9c3423a33dbd
   ```

   For more information on custom jar support, refer Custom Jar Support.
6. Check the right hand pane to view **What I've selected.** If you are happy with the details entered, click **Add**.

Finish up

On the **Finish Up** step of the workflow, you are asked to download the agent you will use to interface between Oracle Access Governance and MySQL database. Select the **Download** link to download the agent zip file to the environment in which the agent will run.

After downloading the agent, follow the instructions explained in the Agent Administration article.

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

For MySQL Database system to integrate with Oracle Access Governance, a driver jar needs to be registered with the agent. For detailed instruction, refer Custom Jar Support.

The Active Directory connector integrates Oracle Access Governance with Microsoft Active Directory. You can establish a connection between Microsoft Active Directory and Oracle Access Governance by entering connection details and configuring the connector. To achieve this, use the **Orchestrated Systems** functionality available in the Oracle Access Governance Console.

Prerequisites

Before you setup and configure a Microsoft Active Directory orchestrated system, you should consider the following pre-requisites and tasks.

Certified Components

The Microsoft Active Directory system can be any of the following:

- **Microsoft Active Directory**
  - Installed on Microsoft Windows Server 2019, 64-bit platform.
  - Installed on Microsoft Windows Server 2016, 64-bit platform.
  - Installed on Microsoft Windows Server 2012, 64-bit platform.
  - Installed on Microsoft Windows Server 2012 R2, 64-bit platform.
  - Installed on Microsoft Windows Server 2008, both 32-bit and 64-bit platforms.
  - Installed on Microsoft Windows Server 2008 R2, both 32-bit and 64-bit platforms.

Supported Operations

The Microsoft Active Directory orchestrated system supports the following operations:

- Create user
- Delete user
- Reset password
- Add group
- Remove group

Create a User Account for Microsoft Active Directory Orchestrated System Operations

Oracle Access Governance requires a user account to access the Microsoft Active Directory system during service operations. Depending on the system you are using, you can create the user in your target system and assign specific permissions and roles to the user.

For Microsoft Active Directory:

You can use a Microsoft Windows 2008 Server (Domain Controller) administrator account for operations. Alternatively, you can create a user account and assign the minimum required rights to the user account.

To create the Microsoft Active Directory user account for operations:

See Also: Microsoft Active Directory documentation for detailed information about performing this procedure.

1. Create a group (for example, AGGroup) on the system. While creating the group, select Security Group as the group type and Global or Universal as the group scope.

> **Note:**
>
> In a parent-child domain setup, create the group in the parent domain.

2. Make this group a member of the Account Operators group.
3. Assign all read permissions to this group. If there are multiple child domains in the forest, then log in to each child domain and add the above group to the Account Operators group of each child domain.

> **Note:**
>
> You assign read permissions on the Security tab of the Properties dialog box for the user account. This tab is displayed only in Advanced Features view. To switch to this view, select Advanced Features from the View menu on the Microsoft Active Directory console.

4. Create a user (for example, AGUser) on the target system. In a parent-child domain setup, create the user in the parent domain.
5. Make the user a member of the group (for example, OIMGroup) created in Step 1.

Configure

You can establish a connection between Microsoft Active Directory and Oracle Access Governance by entering connection details and configuring your Microsoft Active Directory environment. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, specify which type of database you would like to onboard.

1. Select **Microsoft Active Directory** and click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

> **Note:**
>
> The Microsoft Active Directory orchestrated system allows you to manage groups in Microsoft Active Directory using the **I want to manage identity collections for this orchestrated system** option. If selected, this checkbox allows you to manage Microsoft Active Directory groups from within Oracle Access Governance. Any changes made to Microsoft Active Directory groups will be reconciled between Oracle Access Governance and the orchestrated system. Similarly, any changes made in Microsoft Active Directory, will be reflected in Oracle Access Governance

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.

- User
- User manager

2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the target Microsoft Active Directory.

1. In the **Host** field, enter the hostname or IP address for the directory you want to integrate with Oracle Access Governance.
2. In the **Domain Name** field, enter the domain, for example `example.com`.
3. In the **Port** field, enter the value of the TCP/IP port number used to communicate with the LDAP server. The default is 636.
4. Enter the distinguished name which you will use to authenticate to the directory, in the **Principal** field. This is the user you created in Create a User Account for Microsoft Active Directory Orchestrated System Operations.
5. Enter the password of the target distinguished name in the **Password** field. Confirm the password in the **Confirm password** field.
6. Enter a base context from which to begin searches for users and groups into the **Base Contexts** field.

7. In the **Failover** field, enter a list of failover servers in the format `<servername>:<port>, <servername>:<port>, ...`, for example `ADExample1:636, ADExample1:636, ...`
8. In the **SSL** field, ensure that the value **true** is selected.
   Following are the steps to configure SSL on agent:
   a. Use JDK to install and run an agent.
   b. As part of agent installation process, copy `cacerts` of JDK used for agent under agent Installation directory.
   c. Import AD cert to above `cacerts` file using the command

      ```
      <%JAVA_HOME%>/bin/keytool -import -alias OIGAD-cert -file <AD-cert-file> -
      keystore <agent-install-dir>/cacerts
      ```

   d. `Config.properties` should include the following:

      ```
      JAVA_OPTS=-Djavax.net.ssl.trustStore=/app/cacerts-
      Djavax.net.ssl.trustStorePassword=changeit
      ```

9. Check the right hand pane to view **What I've selected.** If you are happy with the details entered, select **Add** to create the orchestrated system.

Finish up

On the **Finish Up** step of the workflow, you are asked to download the agent you will use to interface between Oracle Access Governance and Microsoft Active Directory. Select the **Download** link to download the agent zip file to the environment in which the agent will run.

After downloading the agent, follow the instructions explained in the Agent Administration article.

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with a Microsoft Active Directory system.

Integrate with Microsoft Entra ID

Prerequisites

Before you install and configure a Microsoft Entra ID orchestrated system, you should consider the following prerequisites and tasks.

Certified Components

The Microsoft Entra ID system can be any one of the following:

Table 5-51 Certified Components

Component Type	Component
System	Microsoft Entra ID
System API Version	• Microsoft Entra ID • Microsoft Graph API v1.0 • Microsoft Authentication API version v2.0 (OAuth 2.0)

Supported Modes

Microsoft Entra ID orchestrated system supports the following modes:

- **Authoritative Source**
- **Managed System**

Supported Operations

The Microsoft Entra ID orchestrated system supports the following operations on Microsoft Entra ID:

- Create user
- Delete user
- Reset Password
- Assign Roles to a user
- Revoke Roles from a user
- Assign Licences to a user
- Remove Licences from a user
- Assign SecurityGroup to a user
- Remove SecurityGroup from a user
- Assign OfficeGroup to a user
- Remove OfficeGroup from a user

Default Supported Attributes

The Microsoft Entra ID orchestrated system supports the following default attributes. These attributes are mapped depending on the direction of the connection, for example:

- Data being ingested by Oracle Access Governance from Microsoft Entra ID: `User.givenName` will map to `Identity.firstName`
- Data being provisioned into Microsoft Entra ID from Oracle Access Governance: `account.lastName` will map to `User.surname`

Table 5-52 Default Attributes - Authoritative Source

Microsoft Entra ID Entity	Attribute Name On Managed System	Oracle Access Governance Identity Attribute Name	Oracle Access Governance Identity Attribute Display Name
User	id	uid	Unique Id

Table 5-52 (Cont.) Default Attributes - Authoritative Source

Microsoft Entra ID Entity	Attribute Name On Managed System	Oracle Access Governance Identity Attribute Name	Oracle Access Governance Identity Attribute Display Name
	mailNickname	name	Employee user name
	userPrincipalName	email	Email
	givenName	firstName	First name
	surname	lastName	Last name
	displayName	displayName	Name
	usageLocation	usageLocation	Locality name
	manager	managerLogin	Manager
	preferredLanguage	preferredLanguage	Preferred language
	accountEnabled	status	Status

Table 5-53 Default Attributes - Managed System

Microsoft Entra ID Entity	Attribute Name On Managed System	Oracle Access Governance Account Attribute Name	Oracle Access Governance Account Attribute Display Name
User	id	uid	Unique Id
	userPrincipalName	name	User login
	givenName	firstName	First name
	surname	lastName	Last name
	displayName	displayName	Name
	mailNickname	mailNickname	Mail nick name
	usageLocation	usageLocation	Usage location
	city	city	City
	country	country	Country
	manager	managerLogin	Manager
	passwordProfile.forceChangePasswordNextSignIn	forceChangePasswordNextSignIn	Change password on next logon
	preferredLanguage	preferredLanguage	Preferred language
	userType	userType	Employee type
	accountEnabled	status	Status
	password	password	Password

Microsoft Enterprise Application Configuration and Settings

Before you can establish a connection, you need to perform the following tasks in your Microsoft Entra ID Admin Center for the Enterprise application:

1. Create and register an enterprise application that you want to integrate with Oracle Access Governance. For more information, refer Microsoft documentation.
2. Generate a client secret for the application
3. Grant the following delegated and application permissions for the Microsoft Graph API:
   Delegated Permission
   - Directory.ReadWrite.All

- Group.ReadWrite.All
- GroupMember.ReadWrite.All
- User.Read
- User.ReadWrite

Application Permission

- Directory.ReadWrite.All
- Group.ReadWrite.All
- GroupMember.ReadWrite.All
- User.ReadWrite.All
- RoleManagement.ReadWrite.Directory

4. Click the Grant Admin Consent button to provide directory-wide full permissions to perform the related API tasks for an integrated system

For more information, refer to the Microsoft documentation.

Configure

You can establish a connection between Microsoft Entra ID (formerly Azure Active Directory) and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field.

1. Select **Microsoft Entra ID**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

The default value in each case is *Selected*.

4. Click **Next**.

> **Note:**
>
> The Microsoft Entra ID orchestrated system allows you to manage groups in Microsoft Entra ID using the **I want to manage identity collections for this orchestrated system** option. If selected, this checkbox allows you to manage Microsoft Entra ID groups from within Oracle Access Governance. Any changes made to Microsoft Entra ID groups will be reconciled between Oracle Access Governance and the orchestrated system. Similarly, any changes made in Microsoft Entra ID, will be reflected in Oracle Access Governance

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable

- Delete

3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration Settings

On the **Integration Settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to Microsoft Entra ID.

1. In the **Host** field, enter the host name of the machine hosting your Managed System. For example, for the Microsoft Graph API, you may enter *graph.microsoft.com*
2. In the **Port** field, enter the port number at which the system will be accessible. By default, Microsoft Entra ID uses port 443.
3. Enter the URL of the authentication server that validates the client ID and client secret for your Managed System in the **Authentication Server Url** field. For example, to authenticate the application using the OAuth 2.0 API, enter in the following syntax

   ```
   https://login.microsoftonline.com/<Primary Domain or Directory(tenant ID)>/
   oauth2/v2.0/token
   ```

   To know how to fetch your Primary domain or tenant ID, refer Microsoft documentation.
4. Enter the client identifier (a unique string) issued by the authorization server to your client system during the registration process, into the **Client ID** field. The client ID, also known as Application ID, is obtained when registering an application in Microsoft Entra ID. This value identifies your application in the Microsoft identity platform. For more details refer to Microsoft documentation.
5. In the **Client secret** field, enter the secret ID value to authenticate the identity of your system. You need to create a new client secret for your system and enter the value in this field. Only use this value when you are not using private key for authentication.

> **Note:**
>
> You must note or copy this client secret value, as you won't be able access or view it once you leave the page.

For more details refer Microsoft documentation.

6. Enter PEM private key into the **Private key** field, only when you are not using Client secret for authentication.
   For test purposes only, you can generate a self-signed certificate using the following steps:
   a. Create an encrypted private key which you will load into the Entra ID instance.

      ```
      openssl req -x509 -newkey rsa:2048 -keyout encrypted_key.pem -out
      cert.cer -sha256 -days 365
      ```

   b. Decrypt the private key to create a .pem (decrypted_key.pem in the example) file which you can enter as the value for the **Private key** when configuring Oracle Access Governance.

      ```
      openssl rsa -in encrypted_key.pem -out decrypted_key.pem
      ```

   c. Optionally, if your private key is in PKCS1 format, convert the decrypted key for PKCS8 format which is supported in Oracle Access Governance.

      ```
      openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in
      decrypted_key.pem -out pkcs8.key
      ```

7. Enter the value for the certificate fingerprint (X509) in the **Certificate fingerprint**, only when you are not using Client secret for authentication. .
   To obtain the certificate fingerprint use the following steps:
   a. Convert the hex value of the certificate thumbprint to binary.

      ```
      echo -n "***353DB6DF03567473E299DB5E7F4C***" | xxd -r -p >
      thumbprint.bin
      ```

   b. Convert the binary thumbprint to base64 which can be used in the **Certificate fingerprint** field.

      ```
      openssl base64 -in thumbprint.bin -out thumbprint_base64.txt
      ```

8. Click **Add** to create the orchestrated system.

Finish up

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no postinstall steps associated with a Microsoft Entra ID system.

Prerequisites

Before you install and configure a Microsoft Teams orchestrated system, you should consider the following prerequisites and tasks.

Certified Components

The system must be the following:

- Microsoft Teams

Supported Modes

Microsoft Teams orchestrated system supports the following modes:

- **Managed System**

Supported Operations

The Microsoft Teams orchestrated system supports the following operations:

- Create User
- Delete User
- Reset Password
- Add Teams Group
- Remove Teams Group

Microsoft Teams Application Configuration and Settings

Before you can establish a connection, you need to perform the following tasks in your Microsoft Teams application:

- Create and register an enterprise application that you want to integrate with Oracle Access Governance. For more information, refer Microsoft documentation.
- Generate a client secret for the application
- Assign the following delegated permissions that the client application requires on Microsoft Teams Directory:
  Delegated Permission
  - Read and write directory data
  - Read and write all groups
  - Read all groups
  - Access the directory as the signed-in user
  - Read directory data
  - Read all user's full profiles

  - Read all user's basic profiles
  - Sign in and read user profile
- Add the client application to "Company Administrator" and "User Account Administrator" in the Microsoft Teams administrative roles.
  For more information, refer Microsoft documentation.

Configure

You can establish a connection between Microsoft Teams and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration** → **Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard.

1. Select **Microsoft Teams**.
2. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to Microsoft Teams.

1. In the **Host** field, enter the host name of the machine hosting your Microsoft Teams system.
   For example, for the Microsoft Graph API, you may enter graph.microsoft.com
2. In the **Port** field, enter the port number at which the system will be accessible.

   > **Note:**
   >
   > This field is not mandatory.

3. Enter the **client ID** (a unique string) value. The client ID, also known as Application ID, is obtained when registering an application on Microsoft Entra ID (formerly Azure Active Directory). This value identifies your application in the Microsoft identity platform. For more details refer Microsoft documentation.
4. In the **Client secret** field, enter the secret ID value to authenticate the identity of your client application. You need to create a new client secret for your application and enter the value in this field.
5. In the **Authentication Server Url** field, enter the URL of the authentication server that validates the client ID and client secret for your target system in the Authentication Server Url field.
6. Click **Add** to create the orchestrated system.

Postconfiguration

There are no postinstall steps associated with a Microsoft Teams system.

Prerequisites

Before you install and configure a Flat File orchestrated system, you should consider the following prerequisites and tasks.

Certified Components

The system must be the following:

- **CSV Flat file** located in Oracle Cloud Infrastructure (OCI) Object Storage in your tenancy

Supported Modes

Flat File orchestrated system supports the following modes:

- **Authoritative Source**
- **Managed System**

Supported Operations

The Flat File orchestrated system supports the following operations:

- Create Account
- Delete Account
- Add Entitlement
- Remove Entitlement

Create a bucket in the OCI Object Storage service for Flat File Orchestrated System Operations

In order to load a flat file into Oracle Access Governance you need to place the data files in a bucket created using the OCI Object Storage service. This bucket can be created in any compartment of your OCI tenancy. For details regarding OCI Object Storage, refer to Managing Buckets.

In order to access the bucket, you need to create a service user that has read, write, and delete access (**manage** privileges) to the bucket. Follow this process to create this service user:

- Create a compartment, *accessgovernance*/
- Create a local identity user, *agcs_user* in any domain in your tenancy.
- Create an identity group, *agcs_flatfilegroup* in any domain in your tenancy.
- Assign the identity user *agcs_user* to the identity group *agcs_flatfilegroup*.
- Create a policy, *agcs_flatfilepolicy*, with the following policy statement:

```
allow group <groupname> to manage objects in compartment <compartmentname>
    where target.bucket.name = 'bucketname'
```

For example:

```
allow group agcs_flatfilegroup to manage objects in compartment
accessgovernance
    where target.bucket.name = 'bucket-20231130-1143'
```

Configure

You can establish a connection between Flat File and Oracle Access Governance by entering connection details. To achieve this, use the Orchestrated Systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard.

1. Select **Flat File**.
2. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add Owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

On the **Integration settings** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the Flat File.

1. In the **What is the OCI user's OCID?** field, add the OCID for the OCI user owning the bucket containing the flat files you want to integrate.
2. In the **What is the fingerprint of the OCI user's API key?**. Enter the fingerprint for the OCU user's API key. Consult Required Keys and OCIDs in the OCI documentation for details on how to obtain the value for this.
3. Enter the user's private API key, in PEM format into the **What is the OCI user's private API key in PEM format?** field. Consult Required Keys and OCIDs in the OCI documentation for details on how to obtain the value for this.
4. Enter the tenancy into the **What is the OCI tenancy of the OCI user?** field.
5. Enter the home region code of the tenancy into the **What us the OCI tenancy's home region code?** field. Details of region codes can be found in Regions and Availability Domains OCI documentation.
6. Enter the bucket namespace of the tenancy in the **What is the namespace for the bucket?** field.
7. In the **Enter the name of the bucket where your flat file is stored in OCI object storage** field, enter the name of the bucket where your flat file is stored in OCI object storage.
8. Enter the encoding into the **Encoding** field. Default is UTF-8.
9. In the **Field Delimiter** field, enter the field delimiter character used in the Flat File. Default is `,`.
10. In the **Sub Field Delimiter** field, enter the sub field delimiter character used in the Flat File. Default is `#`.
11. In the **MultiValue Delimiter** field, enter the multivalue delimiter character used in the Flat File. Default is `;`.
12. In the **Text Qualifier** field, enter the character used in the Flat File to act as a text qualifier. Default is `"`.
13. In the **Date Format** field, enter the Java data format in which date type fields are included in the Flat File, for example `dd/MM/yyyy`. If no date format is specified, the date field will be assumed to be of data type Long.
14. If you want to check the connectivity to your Flat File, click the **Test integration** button.
15. Click **Add** to create the orchestrated system.

Finish up

Finally, you are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

Check Bucket Folder Structure

After creation of the orchestrated system, the following folder structure should be created in the defined bucket.

```
<ServiceInstanceName>/<OrchestratedSystemName>
    failed
    inbox
    outbox
    sample
    schema
```

These folders fulfill the following purposes:

- **`failed`**: Files with any kind of data issue will be moved to this folder under the respective entity folder, in the event of a dataload operation failure.
- **`inbox`**: Contains the entity subfolders in which CSV files should be placed to be included in the dataload operation.
- **`outbox`**: Used to output the provisioning events for each entity.
- **`sample`**: Contains example CSVs with the expected header. These can be used as a reference for generating data and putting in the inbox for data load. These files should not be altered.
- **`schema`**: contains the JSON representation of each entity's schema.This can be referred to for understanding details like:
  - `dataType`
  - Mandatory attributes
  - Whether an attribute is multivalued or not
  - If the attribute is complex and has nested attributes (dataType will be CUSTOM)
  - Supported `dataTypes` are:
    - TEXT
    - NUMBER
    - DECIMAL_NUMBER
    - DATE
    - FLAG
    - CUSTOM

Define Custom Attributes

Custom attributes are supported for the `IDENTITY` entity. If you want to include custom attributes in your dataload then you need to add them in the `<ServiceInstanceName>/<OrchestratedSystemName>/schema/IDENTITY.json` file.

Custom attribute names should meet the following requirements:

- start with a character: A-Z or a-z

- contain only characters or numbers: A-Z or a-z or 0-9
- For the DATE type attribute, only long value is supported
- Custom attributes can only be added, they cannot be deleted
- A custom attribute of CUSTOM type cannot be added

Once you have added any custom attributes in the `IDENTITY.json` file, you will need to include them in Oracle Access Governance as described in Fetch Latest Custom Attributes. Once this is completed, the sample CSV will be updated with the newly added custom attribute(s). Update the data files in the inbox to include the custom attribute(s) in your next data load.

Run Dataload

Dataload is on demand. You should run the dataload when you have defined any custom attributes, and have added the relevant CSV data files into the `inbox` folder. Each time you run a dataload it is a full data load, there is no incremental load. UTF-8-BOM encoding is not supported.

If there is any kind of failure (single record or complete file failure), the data load operation will be marked as failed. The files that have been processed successfully will stay in the `inbox` while the failed files will be moved to the `failed` folder. After fixing the data issue, the customer is expected to put the files back in the `inbox` again and retry the dataload. Data integrity issues, such as a permission being assigned to an account that is missing in the CSV), can cause the dataload operation to fail. However, in such cases the CSV files will not be moved to the `failed` folder. Files will be moved to the `failed` folder only when there are issues reading the data itself, such as missing mandatory data.

Generic REST

Generic REST Orchestrated System Overview

The Generic REST Orchestrated System provides a solution to integrate Oracle Access Governance with REST-based identity-aware systems. A REST-based identity-aware system is any system that exposes its REST APIs or interfaces for identity management.

The Generic REST Orchestrated System provides features including the following:

- Full/incremental data load for Authoritative Sources or Managed Systems
- Real-time provisioning
- Cloud native serverless function integration to define REST-based identity-aware system schema, request, response, and test templates

The Generic REST Orchestrated System differs from others in that definitions for schema, request, and response are not fixed. Other Orchestrated Systems have schema, request, response, and test templates pre-loaded for the Authoritative Source or Managed System to which they apply. Since Generic REST Orchestrated Systems can apply to any REST-based identity-aware system, the schema, request, response, and test templates are loaded at runtime, rather than when the Orchestrated System is created.

For each Authoritative Source or Managed System, you will need to create the following templates:

- **grc-schema-template**: This template defines the schema for the Authoritative Source or Managed System you want to integrate.
- **grc-request-template**: This template defines the request format (headers, url, request parameters, request body) required to invoke the Authoritative Source or Managed System API to request identity data.
- **grc-response-template**: This template defines the response format for identity and account data.
- **grc-test-template**: This template defines an API to test the connectivity between Oracle Access Governance and the Authoritative Source or Managed System.

When an operation is invoked the following parameters are passed to OCI Functions.

- Orchestrated system name
- Entity name (identity or account)
- Operation name

The OCI Function is called and returns a JSON file with the templates relevant to the Orchestrated System.

Prerequisites

Before you install and configure a Generic REST Orchestrated System, you should consider the following prerequisites and tasks.

Certified Components

The Managed System can be any one of the following:

- Any identity-aware system that supports REST services

Supported Modes

Generic REST Orchestrated System supports the following configuration modes:

- **Authoritative Source**
- **Managed System**

Use Cases Supported by the Generic REST Orchestrated System

A Generic REST Orchestrated System can be used to on-board identity data into Oracle Access Governance from a REST service, and then efficiently manage identities in an integrated cycle with the rest of the identity-aware systems in your enterprise.

As a business use case example, consider a leading logistics company that has 20+ cloud applications. Most of these cloud applications are now inefficient because data in these applications are manually entered and are managed using spreadsheets or custom-coded process flows. Therefore, this company wants to integrate its cloud applications with Oracle Access Governance to streamline its operations, increase its organizational efficiency, and at the same time, lower its operational costs. There are two approaches for integrating these cloud applications with Oracle Access Governance. One approach would be to deploy a point-to-point connector for each of these applications. The drawbacks of this approach are as follows:

- Increased time and effort to identify and deploy a point-to-point connector for each application.

- Increased administration and maintenance overheads for managing connectors for each application.
- Unavailability of point-to-point connectors for all applications. In such a scenario, one needs to develop custom connectors which increases time and effort to develop, deploy and test the custom connector.

An alternative to this approach is to use the Generic REST Orchestrated System to integrate all the cloud applications with Oracle Access Governance. The Generic REST Orchestrated System provides the ability to manage accounts across all cloud applications without spending additional resources and time on building custom connectors for each cloud application.

The Generic REST Orchestrated System helps enterprises leverage Oracle Access Governance to integrate with Managed Systems for identity governance. These Managed Systems include any application that exposes REST APIs such as SaaS, PaaS, home-grown applications and so on.

The following are some example scenarios in which the Generic REST Orchestrated System is used:

- **User Management**

  The Generic REST Orchestrated System allows you to manage individuals who can access resources by defining them as identities in Oracle Access Governance and assigning them to identity collections and roles. Identities are created from any authoritative Orchestrated System such as Generic REST, on data load.

- **Access Control**

  The Generic REST Orchestrated System manages access control via identity collections, roles, access bundles, and policies. Depending on the orchestrated system being used, you can manage access using Oracle Access Governance self service features, specifically Request Access. For example, you can use the Generic REST Orchestrated System to automatically assign or revoke access to a system based on predefined access policies in Oracle Access Governance. As new users are added to a specific role, they automatically gain corresponding access in the systems covered by the access policy.

Setup OCI Serverless Function to Connect with REST-based Identity Aware System

The Generic REST Orchestrated System requires support from OCI Serverless Functions in order to connect to REST-based identity aware systems.

To setup OCI Functions for use with the Generic Rest Orchestrated System refer to Setup OCI Serverless Function to Connect with REST-based Identity Aware System.

Configure

You can establish an integration between REST-based identity-aware systems and Oracle Access Governance by entering details of the OCI Functions and templates to integrate the REST-based system. To achieve this, use the Orchestrated System functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.

2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to onboard. You can search for the required system by name using the **Search** field. Select the **Generic REST Connector** tile. When you select this tile, a dialog page is shown outlining the steps to configure the Orchestrated System. This includes a link to a sample implementation of the OCI Functions required to connect to REST-based identity aware systems. If you have not done so, you should download the *idm-agcs-generic-rest-reference-implementation.zip* file and develop your own OCI Functions based on this example. For further details on the sample implementation see Setup Sample Implementation. For further details on how to develop the OCI Functions required see Setup OCI Serverless Function to Connect with REST-based Identity Aware System and Generic Rest Schema Discovery.

Once selected, a value of **Generic REST Connector** is displayed on the right hand side under **What I've selected**. Click **Next**.

Enter details

On the **Enter Details** step of the workflow, enter the details for the orchestrated system:

1. Enter a name for the system you want to connect to in the **What do you want to call this system?** field.
2. Enter a description for the system in the **How do you want to describe this system?** field.
3. Determine if this orchestrated system is an authoritative source, and if Oracle Access Governance can manage permissions by setting the following checkboxes.
   - **This is the authoritative source for my identities**
   - **I want to manage permissions for this system**

   The default value in each case is *Selected*.
4. Click **Next**.

Add owners

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager
2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Configure

On the **Configure** step of the workflow, enter the configuration details required to allow Oracle Access Governance to connect to the system using the Generic REST Connector.

1. **What is the OCI user's OCID?**: Enter the Oracle Cloud Identifier (OCID) for the OCI user you will use to connect to the system. For further information regarding OCIDs see Oracle Cloud Identifier, OCID Syntax, and Where to Get the Tenancy's OCID and User's OCID.

For example,
`ocid1.user.oc1..aabdgsegsccawmw2o6qraopae7egmlochlopclhnwxq6pctu6oocgn`

2. **What is the fingerprint for the OCI user's API Key?**: Enter the fingerprint of the public key of the API Signing Key for the OCI instance you will be connecting to. Steps to retrieve the fingerprint can be found in How to Get the Key's Fingerprint, The fingerprint will look similar to this: `12:34:56:78:90:ab:cd:ef:12:34:56:78:90:ab:cd:ef`.
3. **What is the OCI user's private API Key in PEM format?**: Enter the private SSH key (.pem file) for the API Signing Key. Copy it directly from the text editor or use the *cat* command to open the SSH key file from console.
4. **What is the OCI tenancy of the OCI user?**: Enter the OCID for the target tenancy. For further information regarding OCIDs see Oracle Cloud Identifier, OCID Syntax, and Where to Get the Tenancy's OCID and User's OCID.
5. **What is the OCI function's region code?**: Enter the home region for the target OCI tenancy, using the region identifier. The region identifier for your home region can be found in Regions, the identifier for US East (Ashburn) is `us-ashburn-1`, for example. For further information on home region, see The Home Region, and How do I find my tenancy home region?.
6. **What is the OCI function's compartment Id?**: Enter the compartment ID for the function you want to integrate.
7. **What is the OCI function's application name?**: Enter the application name of the function you want to integrate.
8. **Function Version**: Enter the function version of the function you want to integrate.
9. **Request Template Cache TTL In Minutes**: Duration for which the request template will be cached. If time is set as 0, no caching will be done. When the cache expires the OCI function will be invoked to get the new template. The cache time should be less than the token expiry time to avoid dropped connections due to expired token.
10. **Response Template Cache TTL In Minutes**: Duration for which the response template will be cached. If time is set as 0, no caching will be done. When the cache expires the OCI function will be invoked to get the new template. The cache time should be less than the token expiry time to avoid dropped connections due to expired token.
11. **Test Template Cache TTL In Minutes**: Duration for which the test template will be cached. If time is set as 0, no caching will be done. When the cache expires the OCI function will be invoked to get the new template. The cache time should be less than the token expiry time to avoid dropped connections due to expired token.
12. **Schema Template CacheTTL In Minutes**: Duration for which the schema template will be cached. If time is set as 0, no caching will be done. When the cache expires the OCI function will be invoked to get the new template. The cache time should be less than the token expiry time to avoid dropped connections due to expired token.
13. **Read Response Timeout In Seconds**: Enter an integer value that specifies the number of seconds within which response must be received from the orchestrated system
14. **Connect Timeout In Seconds**: An integer value that specifies the number of seconds after which an attempt to establish the connection between the orchestrated system and Oracle Access Governance times out.
15. Click **Add** to create the orchestrated system.

Finish up

On the final step, **Finish up**, you are given a choice whether to further configure your Orchestrated System before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

Once you have configured your Generic REST Orchestrated System, you can navigate to the Orchestrated System page and check operations in the activity log. Some of the operations that you may see include:

- **Schema Discovery**: The Generic REST Orchestrated System is schema-less at design and deployment time. As part of the orchestration lifecycle, schema discovery must take place to update the Orchestrated System with details of the schema and object classes for the required Authoritative Source or Managed System. For details regarding Schema Discovery see Generic Rest Schema Discovery.
- **Validate**: This operation performs the following tasks:
  - Invokes the test template, which in turn invokes the endpoint specified in the template and checks connectivity with the Managed System.
  - Invokes the schema template and retrieves all the schema information for the Managed System including entities and attributes.
- **Lookup Data Load**: If any lookups are defined, the data corresponding to the lookups is loaded.
- **Full Data Load**: This operation will load the data for any entities specified and ingest.

How Does the Generic REST Orchestrated System Use OCI Functions?

Oracle Access Governance leverages OCI Functions for REST API integrations. Each Generic REST Orchestrated System you create is associated with an OCI function which contains the logic to process requests and generate responses. It calls REST APIs of the Managed System for data reconciliation and provisioning operations when triggered by Oracle Access Governance.

For full details of how to use OCI Functions refer to Oracle Cloud Infrastructure Functions.

Configuring Your Tenancy to Enable OCI Functions

Before you can use OCI Functions to create and deploy functions, you must create Oracle Cloud Infrastructure resources to support this. Resources you need to create include a user account, a group to which the user account will belong, a compartment, virtual cloud network (VCN), vaults and secrets, and policies to give the group (and the user accounts that belong to it) access to function-related resources. If suitable resources already exist, there is no need to create new ones.

Create Users and Groups

If you do not have a suitable user or group for managing your OCI functions:

1. Sign on to the Oracle Cloud Infrastructure Console as a tenancy administrator.
2. Open the navigation menu and select **Identity & Security** → **Identity** → **Domains** → **<YourDomain> -> Groups** → **Create Group**.
3. Create a group, for example *ocifn_grp*: For details on how to create a group see To create a group.
4. Select **Identity & Security** → **Identity** → **Domains** → **<YourDomain>** → **Users** → **Create user**.
5. Create a user, for example *ocifn_user*: For details on how to create a user see To create a user.
6. Select **Groups** → **Assign user to groups**.
7. Assign *ocifn_user* to the group: For details on how to add a user to a group see To add a user to a group

Create Compartment

If you do not have a suitable compartment in which you can create network resources and OCI Functions resources:

1. Sign on to the Oracle Cloud Infrastructure Console as a tenancy administrator.
2. Open the navigation menu and select **Identity & Security** → **Identity** → **Compartments**.
3. Select **Create Compartment** and add a compartment, for example *ocifn_compartment*: For details on creating a compartment see To create a compartment

> **Note:**
>
> You create a compartment to own:
>
> - Network resources
> - Function-related resources
>
> You can create a single compartment to contain both sets of resources, or you can create separate compartments for network resources and function-related resources.

Create VCN and Subnets

If you do not have a suitable VCN in which to create network resources:

1. Sign on to the Oracle Cloud Infrastructure Console as a tenancy administrator.
2. Open the navigation menu and select **Networking** → **Virtual cloud networks**.
3. Select **Start VCN Wizard** to create a new VCN.
4. In the dialog, select **Create VCN with Internet Connectivity** and click the **Start VCN Wizard**.
5. In the wizard, enter the following details:
   a. **VCN Name**: Enter the VCN name, for example *ocifn_vcn*.
   b. **Compartment** : Enter the compartment within which the VCN resources will be created, for example *ocifn_compartment*.
6. Click **Next**, then click **Create** to create the VCN and related network resources.

For details of all options when creating the VCN see Creating the VCN and Subnets to Use with OCI Functions.

Create OCI Vault and Secret

If you do not have a suitable vault in which to store client credentials:

1. Sign on to the Oracle Cloud Infrastructure Console as a tenancy administrator.
2. Open the navigation menu and select **Identity & Security → Key Management & Secret Management**
3. Click **Create Vault**.
4. Select the compartment into which the Vault will be created, and give the Vault a Name. Click **Create Vault**.
5. Open the Vault you created and click **Create Key**. Create the Key in the same compartment you created your Vault. Give the Key a name and click **Create Key**.
6. From the **Resources** menu, select **Secrets**. Click **Create Secret**.
7. On the **Create Secret** page, add details for compartment, name, and description. Select the key you created from the **Encryption Key** drop down list. Select the **Manual secret generation** option, and enter your client code and client secret in one of the following formats into the **Secret Contents** field.
   a. **OAuth-based Authorization**

   ```
   {
     "clientCode": "69b48365-14e2-430a-bd75-171f89c158fa",
     "clientSecret": "*/Z9g:gM*SWWLLCGkQhQg2hdWelAZD82"
   }
   ```

   For sample code to create authorization token for OAuth-based authorization, see OAuth Authorization - Sample Token Creation Code.

   b. **Basic Authorization**

   ```
   {
     "username": "69b48365-14e2-430a-bd75-171f89c158fa",
     "password": "*/Z9g:gM*SWWLLCGkQhQg2hdWelAZD82"
   }
   ```

   .

   For sample code to create authorization token for Basic authorization, see Basic Authorization - Sample Token Creation Code
8. Select **Create Secret** to save your changes.

The secret you have created can be used to pass client secrets in your request function. Select the secret you created and copy the value of the secret OCID, which will look something like `ocid1.vaultsecret.oc1.iad.dyyyyehdl4ggaawnqt47hfj48ltofzkdg6wy5fjne859jg0`.

Edit the `<SampleBase>/grc-request-template/src/main/resources/request/applications/<YourApplicationName>/config.yaml` file and add the details of the secret and region, for example:

```
secretId:
"ocid1.vaultsecret.oc1.iad.dyyyyehdl4ggaawnqt47hfj48ltofzkdg6wy5fjne859jg0"
region: "us-ashburn-1"
```

For further details on OCI Vault Secrets see Managing Vault Secrets.

Create or Update your Dynamic Group

In order to use OCI Vault and Secret Services, your function must be part of a dynamic group. For details on how to create your dynamic group, follow the instructions given in Managing Dynamic Groups.

As part of the creation of your dynamic group you will be required to define a set of *matching rules* to define the group member. When specifying the matching rule it is recommended that you match all functions in a compartment with the following rule:

```
ALL { resource.type = 'fnfunc', resource.compartment.id =
'ocid1.compartment.oc1..fdfdfege4om6nat7fue56566556qqvj3eesjqhmjaegeiaxa' }
```

Create Policy for OCI Vault and Secret

Create a new policy that allows the dynamic group created in above step to read secret-family in your tenancy. For further information on creating OCI policies see Policy Syntax

1. Sign on to the Oracle Cloud Infrastructure Console as a tenancy administrator.
2. Open the navigation menu and select **Identity & Security → Identity → Policies**.
3. Select **Create Policy**, specify a name for the policy, for example *ocifn_policy*, and select the tenancy's root compartment.
4. Add a policy statement similar to:

   ```
   Allow dynamic-group <dynamic-group-name> to read secret-family in tenancy
   ```

Create Policy for Group and Service

If one or more OCI Functions users, for example *ocifn_user* is not a tenancy administrator:

1. Sign on to the Oracle Cloud Infrastructure Console as a tenancy administrator.
2. Open the navigation menu and select **Identity & Security → Identity → Policies**.
3. Select **Create Policy**, specify a name for the policy, for example *ocifn_policy*, and select the tenancy's root compartment.
4. Using Policy Builder, select **Functions** from the **Policy use cases** list, and base your policy on the **Let users create, deploy, and manage functions and applications** template.
5. Select the group created earlier, *ocifn_group* and the *ocifn_compartment* for the location. This should give you a policy statement similar to:

   ```
   Allow group 'Default'/'ocifn_grp' to use cloud-shell in tenancy
   Allow group 'Default'/'ocifn_grp' to manage repos in compartment
   ```

```
ocifn_compartment
Allow group 'Default'/'ocifn_grp' to read objectstorage-namespaces in
tenancy
Allow group 'Default'/'ocifn_grp' to manage logging-family in compartment
ocifn_compartment
Allow group 'Default'/'ocifn_grp' to read metrics in compartment
ocifn_compartment
Allow group 'Default'/'ocifn_grp' to manage functions-family in
compartment ocifn_compartment
Allow group 'Default'/'ocifn_grp' to use virtual-network-family in
compartment ocifn_compartment
Allow group 'Default'/'ocifn_grp' to use apm-domains in compartment
ocifn_compartment
Allow group 'Default'/'ocifn_grp' to read vaults in compartment
ocifn_compartment
Allow group 'Default'/'ocifn_grp' to use keys in compartment
ocifn_compartment
Allow service faas to use apm-domains in compartment ocifn_compartment
Allow service faas to read repos in tenancy where
request.operation='ListContainerImageSignatures'
Allow service faas to {KEY_READ} in compartment ocifn_compartment where
request.operation='GetKeyVersion'
Allow service faas to {KEY_VERIFY} in compartment ocifn_compartment where
request.operation='Verify'
```

For more details on creating policies for OCI Functions see Creating Policies to Control Access to Network and Function-Related Resources

Create an Application to Group Your OCI Functions

In OCI Functions, an application is a logical grouping of functions. The properties you specify for an application determine resource allocation and configuration for all functions in that application. You have to create a function within an application, so at least one application must exist before you can create a function in OCI Functions.

To create an application:

1. Sign on to the Oracle Cloud Infrastructure Console as a functions developer.
2. Open the navigation menu and select **Developer Services** → **Functions** → **Applications**.
3. Select the compartment you are using for Generic REST orchestrated system functions.
4. Click **Create application**.
5. Add a name for the application, for example *agcs-generic-rest-connector*. You will deploy your functions into this application, and specify this application when invoking the function.
6. Select the VCN and subnet in which to run the function.
7. Click **Create**.

For full details on creating applications with OCI Functions see Creating Applications.

Setup Environment for Developing OCI Functions

In order to create and deploy your OCI Functions you need to install and setup a development environment.

Before you can create OCI functions for the Generic REST Orchestrated System. you need to install and configure a development environment, if you do not already have one available. Setup of a development environment can be done in a number of ways, and includes some or all of the following steps:

- Install and start Docker
- Setup API signing key and OCI profile
- Install Fn Project command line interface (CLI)
- Setup Fn Project CLI context provider
- Complete Fn Project CLI context configuration
- Generate auth token
- Log into registry

Full details of how to setup your development environment in different in different contexts can be found in the Oracle Cloud Infrastructure documentation as follows:

- Cloud Shell
- Local Host
- OCI Compute Instance

Once your development environment is setup, you can start to develop and deploy functions for the Generic REST Orchestrated System.

Develop and Deploy Function to OCI

In order to deploy the Generic REST Orchestrated System you need to develop and deploy an OCI function which enables you to connect to the identity-aware system that you want to integrate with Oracle Access Governance using REST.

To develop and deploy OCI Functions for use with the Generic REST Orchestrated System:

Setup Sample Implementation

To help with development of your OCI Functions a sample implementation is provided with the Generic REST Orchestrated System. To download this example you should follow the instructions given in the Select System procedure.

The sample implementation downloaded, *idm-agcs-generic-rest-reference-implementation.zip*, comprises four use cases. When unzipping the file, you will see the following four directories from which you can choose a sample that meets your requirements.

> **Note:**
>
> In the details that follow, the directory to which you have unzipped the sample implementation is referred to as `<SampleBase>`. Use this to locate the files mentioned. For example:
> If you unzip the sample implementation to a directory, say, `/myagfunctions` then the file `<SampleBase>/grc-schema-template/src/main/resources/schema/config.yaml` can be found at `/myagfunctions/grc-schema-template/src/main/resources/schema/config.yaml`.

1. **idm-agcs-serverless-getting-started-sample**: This is a basic and easy-to-understand sample with minimal code, demonstrating how functions work. This use case is ideal for customers needing to integrate a single application with the Generic REST connector, requiring minimal configuration. Initial set up is with an Oracle Identity Cloud Service (IDCS) application. Detailed steps to configure and deploy this function can be found in the `README.md` file within this directory.
2. **idm-agcs-serverless-idcs-application-sample**: This sample focuses on the IDCS application example with minimal configuration. It includes the framework to add more applications in the future within this function. Configuration and deployment steps are detailed in the `README.md` file located in this directory.
3. **idm-agcs-serverless-azuread-application-sample**: This sample is specifically for customers who need an AzureAD application example with minimal configuration. It contains the framework to add more applications in the future within this function. Detailed steps to configure and deploy this function are available in the `README.md` file inside this directory.
4. **idm-agcs-serverless-multi-application-sample**: This sample is for customers needing both IDCS and AzureAD application examples with minimal configuration. It includes the framework to add more applications in the future within this function. The `README.md` file in this directory provides all necessary steps to configure and deploy this function.

Create Functions for Your Application

Create the functions for your own REST-based identity-aware system, by referring to the reference templates, and configuring the functions as defined below, with values specific to your application.

Table 5-54 GRC-SCHEMA-TEMPLATE

Item	Description
Input	Orchestrated System name

Table 5-54 (Cont.) GRC-SCHEMA-TEMPLATE

Item	Description

Configuration	List of configured custom REST identity aware systems:`<SampleBase>/grc-schema-template/src/main/resources/schema/config.yaml`
	For example, from the sample implementation:
	`applications:`
	`- "idcs"`
	List of Orchestrated Systems linked to custom REST identity aware system: `<SampleBase>/grc-schema-template/src/main/resources/schema/applications/<YourApplicationName>/config.yaml`
	For example, from the sample implementation:
	`application: "idcs"`
	`connectedSystemConfigs:`
	`- connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_1>>"`
	`- connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_2>>"`
	`- connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_3>>"`

Output	JSON Schema template document, with one or more entity schemas, including any of the following entity types: • ACCOUNT • ENTITLEMENT • LOOKUP • TARGETACCOUNT `<SampleBase>/grc-schema-template/src/main/resources/schema/applications/<YourApplicationName>/TEMPLATE.json` For an example from the sample implementation see Schema Output.

Table 5-55 GRC-REQUEST-TEMPLATE

Item	Description
Input	• Orchestrated System name • Operation name • Entity name

Table 5-55 (Cont.) GRC-REQUEST-TEMPLATE

Item	Description

Configuration	List of configured custom REST identity aware systems: `<SampleBase>/grc-request-template/src/main/resources/request/config.yaml`
	For example, from the sample implementation:
	
	`applications:`
	`  - "idcs"`
	
	List of Orchestrated Systems linked to custom REST identity aware system: `<SampleBase>/grc-request-template/src/main/resources/request/applications/<YourApplicationName>/config.yaml`
	For example, from the sample implementation:
	
	`application: "idcs"`
	`connectedSystemConfigs:`
	`  - connectedSystemName:`
	`"<<CONNECTED_SYSTEM_NAME_1>>"`
	`    authenticationDetail:`
	`      scheme: "https"`
	`      host:`
	`"<<IDCS_AUTH_API_HOST_1>>"`
	`      path: "/oauth2/v1/token"`
	`      method: "POST"`
	`      grantType: "client_credentials"`
	`      scope:`
	`"urn:opc:idm:__myscopes__"`
	`      secretId: "<<SECRET_ID_1>>"`
	`      region: "<<REGION>>"`
	`    applicationInstanceDetail:`
	`      scheme: "https"`
	`      host: "<<IDCS_API_HOST_1>>"`
	`  - connectedSystemName:`
	`"<<CONNECTED_SYSTEM_NAME_2>>"`
	`    authenticationDetail:`
	`      scheme: "https"`
	`      host:`
	`"<<IDCS_AUTH_API_HOST_2>>"`
	`      path: "/oauth2/v1/token"`
	`      method: "POST"`
	`      grantType: "client_credentials"`
	`      scope:`
	`"urn:opc:idm:__myscopes__"`
	`      secretId: "<<SECRET_ID_2>>"`
	`      region: "<<REGION>>"`
	`    applicationInstanceDetail:`
	`      scheme: "https"`
	`      host: "<<IDCS_API_HOST_2>>"`

Table 5-55 (Cont.) GRC-REQUEST-TEMPLATE

Item	Description

	<pre>- connectedSystemName:
	"<<CONNECTED_SYSTEM_NAME_3>>"
	authenticationDetail:
	scheme: "https"
	host:
	"<<IDCS_AUTH_API_HOST_3>>"
	path: "/oauth2/v1/token"
	method: "POST"
	grantType: "client_credentials"
	scope:
	"urn:opc:idm:__myscopes__"
	secretId: "<<SECRET_ID_3>>"
	region: "<<REGION>>"
	applicationInstanceDetail:
	scheme: "https"
	host: "<<IDCS_API_HOST_3>>"</pre>

Output	JSON Request template document, for defined entities and operations: **Entities** • ACCOUNT • ENTITLEMENT • LOOKUP • TARGETACCOUNT **Operations** • CREATE • GET • SEARCH • DELETE • UPDATE • ADD_CHILD_DATA • REMOVE_CHILD_DATA • ENABLE • DISABLE <SampleBase>/grc-schema-template/src/main/resources/request/applications/<YourApplicationName>/<EntityName>/<Operation>_TEMPLATE.json For an example from the sample implementation see Request Output.

Table 5-56 GRC-RESPONSE-TEMPLATE

Item	Description
Input	• Orchestrated System name • Operation name • Entity name

Table 5-56 (Cont.) GRC-RESPONSE-TEMPLATE

Item	Description

Configuration	List of configured custom REST identity aware systems: `<SampleBase>/grc-response-template/src/main/resources/response/config.yaml`
	For example, from the sample implementation:
	`applications:`
	`- "idcs"`
	List of Orchestrated Systems linked to custom REST identity aware system: `<SampleBase>/grc-response-template/src/main/resources/response/applications/<YourApplicationName>/config.yaml`
	For example, from the sample implementation:
	`application: "idcs"`
	`connectedSystemConfigs:`
	`- connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_1>>"`
	`- connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_2>>"`
	`- connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_3>>"`

Output	JSON Response template document, for defined entities and operations: **Entities** • ACCOUNT • ENTITLEMENT • LOOKUP • TARGETACCOUNT **Operations** • CREATE • GET • SEARCH • DELETE • UPDATE • ADD_CHILD_DATA • REMOVE_CHILD_DATA • ENABLE • DISABLE `<SampleBase>/grc-response-template/src/main/resources/response/applications/<YourApplicationName>/<EntityName>/<Operation>_TEMPLATE.json` For an example from the sample implementation see Response Output.

Table 5-57 GRC-TEST-TEMPLATE

Item	Description
Input	Orchestrated System name

Table 5-57 (Cont.) GRC-TEST-TEMPLATE

Item	Description

Configuration	List of configured custom REST identity aware systems: `<SampleBase>/grc-test-template/src/main/resources/test/config.yaml`
	For example, from the sample implementation:
	
	`applications:`
	`  - "idcs"`
	
	List of Orchestrated Systems linked to custom REST identity aware application: `<SampleBase>/grc-test-template/src/main/resources/test/applications/<YourApplicationName>/config.yaml`
	For example, from the sample implementation:
	
	`application: "idcs"`
	`connectedSystemConfigs:`
	`  - connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_1>>"`
	`    authenticationDetail:`
	`      scheme: "https"`
	`      host: "<<IDCS_AUTH_API_HOST_1>>"`
	`      path: "/oauth2/v1/token"`
	`      method: "POST"`
	`      grantType: "client_credentials"`
	`      scope: "urn:opc:idm:__myscopes__"`
	`      secretId: "<<SECRET_ID_1>>"`
	`      region: "<<REGION>>"`
	`    applicationInstanceDetail:`
	`      scheme: "https"`
	`      host: "<<IDCS_API_HOST_1>>"`
	`  - connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_2>>"`
	`    authenticationDetail:`
	`      scheme: "https"`
	`      host: "<<IDCS_AUTH_API_HOST_2>>"`
	`      path: "/oauth2/v1/token"`
	`      method: "POST"`
	`      grantType: "client_credentials"`
	`      scope: "urn:opc:idm:__myscopes__"`
	`      secretId: "<<SECRET_ID_2>>"`
	`      region: "<<REGION>>"`
	`    applicationInstanceDetail:`
	`      scheme: "https"`
	`      host: "<<IDCS_API_HOST_2>>"`

Table 5-57 (Cont.) GRC-TEST-TEMPLATE

Item	Description
	- connectedSystemName: "<<IDCS_CONNECTED_SYSTEM_NAME_3>>" authenticationDetail: scheme: "https" host: "<<IDCS_AUTH_API_HOST_3>>" path: "/oauth2/v1/token" method: "POST" grantType: "client_credentials" scope: "urn:opc:idm:__myscopes__" secretId: "<<SECRET_ID_3>>" region: "<<REGION>>" applicationInstanceDetail: scheme: "https" host: "<<IDCS_API_HOST_3>>"

Table 5-57 (Cont.) GRC-TEST-TEMPLATE

Item	Description

Output	JSON Test Request template document: `<SampleBase>/grc-test-template/src/main/resources/test/applications/<YourApplicationName>/TEMPLATE.json`
	For example, from the sample implementation:
	
	`{`
	`  "id": "1",`
	`  "name": "Test connectivity invoking schema API",`
	`  "method": "GET",`
	`  "uri":`
	`  {`
	`  "scheme": "<<SCHEME>>",`
	`  "host": "<<HOST>>",`
	`  "path": "/admin/v1/Schemas"`
	`  },`
	`  "queryParameters":`
	`  [`
	`  {`
	`  "name": "startIndex",`
	`  "value": "1"`
	`  },`
	`  {`
	`  "name": "count",`
	`  "value": "1"`
	`  }`
	`  ],`
	`  "headers":`
	`  [`
	`  {`
	`  "name": "Content-Type",`
	`  "value": "application/json"`
	`  },`
	`  {`
	`  "name": "Authorization",`
	`  "value": "<<CREDENTIALS>>"`
	`  }`
	`  ]`
	`}`

Test Infra Module Guide

Request and Response templates are complex and errors can occur during development. It is crucial to identify and validate all configured Request and Response templates for correctness during the build time itself.

To address this challenge, the *grc-test-infra* module has been introduced. This module includes tests corresponding to each configured Request and Response template. These tests aim to verify the requests made to the target application's REST API and the responses received. To

achieve this, the tests utilize a utility, such as a generic Request and Response processor, capable of handling any REST API call developed within the module. Essentially, it internally interacts with the target application's endpoints and validates the responses based on the configured response templates.

To effectively identify and validate all configured GRC Request and Response templates for correctness during the build time, follow these steps:

1. Develop tests following the example outlined below:
   grc-test-infra sample test

```
package com.oracle.idm.agcs.grc.fn.test.infra;

public class RequestResponseTemplateValidationTest {

    public void validateSearchUserAsIdentityRequestResponseTemplate()
throws IOException, URISyntaxException, InterruptedException {
        // Get the configured Request template by passing the
connectedSystem Name, entity name, and operation
        RequestTemplateOutput requestTemplate =
getRequestTemplateOutput(connectedSystemName, UserAsIdentityEntity,
Operation.SEARCH);

        // Get the configured Response template by passing the
connectedSystem Name, entity name, and operation
        ResponseTemplateOutput responseTemplate =
getResponseTemplateOutput(connectedSystemName, UserAsIdentityEntity,
Operation.SEARCH);

        // Pass the required attributes in the request template by putting
them in the attribute map
        HashMap<String, Object> attributesMap = new HashMap<>();

        // Call the generic utility method to validate request and
response template
        Map<String, Object> lastRecord =
processAndValidateRequestAndResponseTemplate(requestTemplate,
responseTemplate, attributesMap);

        // Retrieve the necessary data from the response for further
processing
        userAsIdentityUid = ((ArrayList<String>)
lastRecord.get("uid")).get(0);
    }
}
```

- The first line inside the test is to get the configured Request template by passing the connectedSystem Name, entity name, and operation.
- The second line gets the configured Response template using similar parameters.
- Users need to pass the required attributes in the request template by putting them in the attribute map.
- Then, users just need to call the generic utility method processAndValidateRequestAndResponseTemplate, which internally validates both the request and response templates for correctness by invoking the API and mapping its response with the configured response template.

- The test will pass if the configured request and response templates are correctly configured; otherwise, it will fail, allowing users to see the failure reason and correct the templates accordingly.

The example test file can be found in the sample implementation at the following location: `<SampleBase>/grc-test-infra/src/test/java/com/oracle/idm/agcs/grc/fn/test/infra/RequestResponseTemplateValidationTest.java`

2. Add the following prerequisites before running the test:
   a. Update the `<SampleBase>/grc-test-infra/src/test/resources/config` file with the following values for your environment:

```
[DEFAULT]
user=<<USER>>
fingerprint=<<FINGERPRINT>>
key_file=<<PEM_FILE_RELATIVE_PATH>>
tenancy=<<TENANCY>>
region=<<REGION>>
```

   b. Update the `<SampleBase>/grc-test-infra/src/test/resources/config.properties` file with the following value for your environment:

```
connectedSystemName=<connectedSystemName>
```

   c. Add the private SSH key (.pem file) for the API Signing Key to the following location:

```
<SampleBase>/grc-test-infra/src/test/resources/oci_api_key.pem
```

3. Execute the developed test and build the function:
   a. Navigate to the `functions` directory of the sample implementation.
   b. Compile and package the functions

```
mvn clean package-DisDevMode=true
```

Build and Deploy Your Function

1. Build your function.

```
cd <SampleBase>
mvn package
```

2. Deploy your function on OCI. The steps should be carried out for each function
   - grc-schema-template
   - grc-request-template
   - grc-response-template
   - grc-test-template

```
cd <SampleBase>/grc-schema-template
fn -v deploy --app <ApplicationName>
```

A Generic REST Orchestrated System is created with a minimal template, which is updated at runtime using OCI Functions to update the Orchestrated System with object classes, lookup types, and outbound transformation data.

Schema Discovery High-Level Workflow

Schema discovery is the process by which a Generic REST Orchestrated System applies OCI Function templates to enable discovery of the schema, object classes, attributes, and connection details for the configured Authoritative Source or Managed System.

Schema discovery occurs when an Orchestrated System is created in Oracle Access Governance. Simplified workflows for Day0 and DayN scenarios are detailed in the following tables.

Day0 Workflow

Table 5-58 Day0 Workflow

Step #	Task/Operation	Description
1	Create Orchestrated System	Create Orchestrated System using the Oracle Access Governance Console. The Orchestrated System is created with details of the OCI Function you have created to return details from the required Authoritative Source or Managed System. Based on the details entered into the Orchestrated System, the following operation is created. • **Validate**
2	Validate operation execution	The Validate operation is executed and, based on the configured schema template, fetches the schema, including the following object classes: • **Identity**: fetches core attributes only. Must have the mandatory core objects detailed here • **Non-identity**: fetches all attributes. Must have the minimum attributes detailed here The configure test template is called to validate connectivity with the Managed System.
3	Post Validate operation	On successful execution of the Validate operation, the following operations are created: • **Lookup Data Load** • **Full Data Load**

DayN Workflow

Table 5-59 DayN Workflow

Step #	Task/Operation	Description
1	Select **Fetch Attributes** from the **Identity Attributes** page of the Oracle Access Governance. See Fetch Latest Custom Attributes for further details.	A **Schema Discovery** operation is created in the Orchestrated System
2	Schema Discovery operation execution	The Schema Discovery operation is executed and, based on the configured schema template, fetches the schema, including the following object classes: • **Identity**: fetches core and custom attributes • **Non-identity**: fetches all attributes
3	Refresh the **Identity Attributes** page	The custom attributes list starts showing the new custom attributes for identity object class.

If all the above completes successfully, the Orchestrated System is available for scheduler to create subsequent Full Data Load operations.

Generic REST Schema Discovery Mandatory Schema Attributes

During schema discovery certain mandatory attributes must be returned as part of the schema output.

Identity Object Class

The mandatory attributes that must be returned as part of the schema output for the Identity object class are:

- uid
- name
- email
- firstName
- middleName
- lastName
- displayName
- employeeType
- title
- empNo
- status
- jobCode
- state

- risk
- location
- department
- managerUid
- managerLogin
- organizationUid
- organizationName
- country
- postalCode
- territory

```
[
  {
    "name": "uid",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "name",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "email",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "firstName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "middleName",
```

```
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "lastName",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "displayName",
      "dataType": "TEXT",
      "nature": [
        "REQUIRED"
      ],
      "usage": [
        "READ"
      ]
    },
    {
      "name": "employeeType",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "title",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "empNo",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "status",
      "dataType": "FLAG",
      "usage": [
        "READ"
      ]
    },
    {
      "name": "jobCode",
      "dataType": "TEXT",
      "usage": [
        "READ"
      ]
```

```
},
{
  "name": "state",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
},
{
  "name": "risk",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
},
{
  "name": "location",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
},
{
  "name": "department",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
},
{
  "name": "managerUid",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
},
{
  "name": "managerLogin",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
},
{
  "name": "organizationUid",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
},
{
  "name": "organizationName",
  "dataType": "TEXT",
  "usage": [
    "READ"
  ]
```

```
  },
  {
    "name": "country",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "postalCode",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "territory",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  }
]
```

Non-Identity Object Class

The mandatory attributes that must be returned as part of the schema output for the Non-Identity object class are:

- uid
- name

```
[
  {
    "name": "uid",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "name",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  }
]
```

Schema Template Outline

In order to support schema discovery you need to create your schema template using the supported outline.

Schema Outline

The schema outline you should follow when creating your schema template is:

```
{
 "schemaTemplates":[
    {
        "type": "", // Type of entity i.e. either object class type i.e.
"ACCOUNT", "ENTITLEMENT", "TARGETACCOUNT" or "LOOKUP"
        "name": "", // Name of entity i.e. name of either object class or
lookup
        "displayName": "", // display name of entity
        "data": {
            // Key-value pairs representing lookup data if any, or else it
will be missing from here.
        }
        "attributes": [
            {
                "name": "", // Name of attribute
                "dataType": "", // Either of TEXT, DATE, NUMBER,
DECIMAL_NUMBER, FLAG
                "nature": [ // Adjectives i.e. One or more of "REQUIRED",
"MULTIVALUED", "SENSITIVE". It can be missing from here if nothing applies.
                ],
                "usage": [ // Verbs i.e. One or more of "READ", "PROVISION".
It can be missing from here if nothing applies.
                ],
                "relationship": { // Entity relationship details
                    "relatedTo": "", // Entity name in relationship with
                    "relatedBy": "", // Attribute to define the relation
                    "relationshipProperties": [ // Additional relationship
properties
                        {
                            "name": "", // Name of additional attribute
                            "dataType": "", // Either of TEXT, DATE, NUMBER,
DECIMAL_NUMBER, FLAG
                            "nature": [ // Only READ_ONLY is possible, or
else it will be missing from here
                            ]
                        }
                    ]
                },
                "outboundTransformation": { // Outbound transformation script
if applicable, or it will be missing from here
                    "script": "" // Script to execute for transformation
                },
                "uiProperties": { // ARMD if applicable, or it will be
missing from here
                    "inputType": "" // Either of Auto, User, Admin
                    "widget": "", // Widget to use on UI i.e. Either of Text,
```

```
Password, Number, Date, SelectOne, RepeatableFieldSet, CheckboxSet
                    "title": "", // Title to use on UI
                    "labelHint": "", // Labelhint to use on UI
                    "minLength": {SOME_POSITIVE_NUMBER},
                    "maxLength": {SOME_POSITIVE_NUMBER},
                    "defaultValues": [ // Default values if applicable, or it
will be missing from here
                    ]
                }
            }
        ]
    }
  ]
}
```

Schema Template Output

Output of the schema template is returned as a JSON document.

Schema Template Output

Your schema template output will look similar to that provided in the reference implementation (`<ReferenceBase>/functions/grc-schema-template/src/main/resources/schema/applications/<YourApplicationName>/TEMPLATE.json`):

```
{
  "schemaTemplates": [
    {
      "type": "ACCOUNT",
      "name": "UserAsIdentity",
      "displayName": "User As Identity",
      "attributes": [
        {
          "name": "uid",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED"
          ],
          "usage": [
            "READ"
          ]
        },
        {
          "name": "name",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED"
          ],
          "usage": [
            "READ"
          ]
        },
        {
          "name": "email",
          "dataType": "TEXT",
          "nature": [
```

```
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "firstName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "middleName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "lastName",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "displayName",
    "dataType": "TEXT",
    "nature": [
      "REQUIRED"
    ],
    "usage": [
      "READ"
    ]
  },
  {
    "name": "employeeType",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "title",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "empNo",
    "dataType": "TEXT",
    "usage": [
      "READ"
```

```
    ]
  },
  {
    "name": "jobCode",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "state",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "risk",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "location",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "department",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "managerUid",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "organizationUid",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "organizationName",
    "dataType": "TEXT",
    "usage": [
      "READ"
```

```
    ]
  },
  {
    "name": "postalCode",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "territory",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "usageLocation",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "country",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ],
    "relationship": {
      "relatedTo": "countries",
      "relatedBy": "uid"
    }
  },
  {
    "name": "managerLogin",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "preferredLanguage",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ],
    "relationship": {
      "relatedTo": "languages",
      "relatedBy": "uid"
    }
  },
  {
    "name": "status",
    "dataType": "FLAG",
    "usage": [
```

```
            "READ"
          ]
        }
      ]
    },
    {
      "type": "ENTITLEMENT",
      "name": "GroupAsEntitlement",
      "displayName": "Group As Entitlement",
      "attributes": [
        {
          "name": "uid",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED"
          ],
          "usage": [
            "READ"
          ]
        },
        {
          "name": "name",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED"
          ],
          "usage": [
            "READ"
          ]
        }
      ]
    },
    {
      "type": "TARGETACCOUNT",
      "name": "UserAsAccount",
      "displayName": "User As Account",
      "attributes": [
        {
          "name": "uid",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED"
          ],
          "usage": [
            "READ",
            "PROVISION"
          ],
          "uiProperties": {
            "inputType": "Auto",
            "widget": "Text",
            "title": "User ID",
            "labelHint": "User ID",
            "minLength": 1,
            "maxLength": 256
          }
        },
```

```
{
  "name": "name",
  "dataType": "TEXT",
  "nature": [
    "REQUIRED"
  ],
  "usage": [
    "READ",
    "PROVISION"
  ],
  "outboundTransformation": {
    "script": "user.getPrimaryEmail()"
  },
  "uiProperties": {
    "inputType": "Auto",
    "widget": "Text",
    "title": "User Name",
    "labelHint": "User Name",
    "minLength": 1,
    "maxLength": 256
  }
},
{
  "name": "email",
  "dataType": "TEXT",
  "nature": [
    "REQUIRED"
  ],
  "usage": [
    "READ",
    "PROVISION"
  ],
  "outboundTransformation": {
    "script": "user.getPrimaryEmail()"
  },
  "uiProperties": {
    "inputType": "Auto",
    "widget": "Text",
    "title": "Email",
    "labelHint": "Email",
    "minLength": 1,
    "maxLength": 256
  }
},
{
  "name": "firstName",
  "dataType": "TEXT",
  "usage": [
    "READ",
    "PROVISION"
  ],
  "outboundTransformation": {
    "script": "user.getName().getGivenName()"
  },
  "uiProperties": {
    "inputType": "Auto",
```

```
            "widget": "Text",
            "title": "First Name",
            "labelHint": "First Name",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "lastName",
          "dataType": "TEXT",
          "usage": [
            "READ",
            "PROVISION"
          ],
          "outboundTransformation": {
            "script": "user.getName().getFamilyName()"
          },
          "uiProperties": {
            "inputType": "Auto",
            "widget": "Text",
            "title": "Last Name",
            "labelHint": "Last Name",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "displayName",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED"
          ],
          "usage": [
            "READ",
            "PROVISION"
          ],
          "outboundTransformation": {
            "script": "[user.getName().getGivenName(),
user.getName().getFamilyName()].filter(element => { return element !== null
&& element.length > 0}).join(' ')"
          },
          "uiProperties": {
            "inputType": "Auto",
            "widget": "Text",
            "title": "Display Name",
            "labelHint": "Display Name",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "mailNickname",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED"
          ],
```

```
          "usage": [
            "READ",
            "PROVISION"
          ],
          "outboundTransformation": {
            "script": "[user.getName().getGivenName(),
user.getName().getFamilyName()].filter(element => { return element !== null
&& element.length > 0}).join('')"
          },
          "uiProperties": {
            "inputType": "Auto",
            "widget": "Text",
            "title": "Nick Name",
            "labelHint": "Nick Name",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "password",
          "dataType": "TEXT",
          "nature": [
            "REQUIRED",
            "SENSITIVE"
          ],
          "usage": [
            "READ",
            "PROVISION"
          ],
          "uiProperties": {
            "inputType": "User",
            "widget": "Password",
            "title": "Password",
            "labelHint": "Password",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "usageLocation",
          "dataType": "TEXT",
          "usage": [
            "READ",
            "PROVISION"
          ],
          "relationship": {
            "relatedTo": "countries"
          },
          "outboundTransformation": {
            "script": "user.getLocation() != null ?
transformationUtil.getLookupCode(agcs_tenant_id, agcs_target_id, 'countries',
user.getLocation()) : null"
          },
          "uiProperties": {
            "inputType": "Auto",
            "widget": "SelectOne",
```

```
            "title": "Location",
            "labelHint": "Location",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "city",
          "dataType": "TEXT",
          "usage": [
            "READ",
            "PROVISION"
          ],
          "uiProperties": {
            "inputType": "User",
            "widget": "Text",
            "title": "City",
            "labelHint": "City",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "country",
          "dataType": "TEXT",
          "usage": [
            "READ",
            "PROVISION"
          ],
          "relationship": {
            "relatedTo": "countries",
            "relatedBy": "uid"
          },
          "outboundTransformation": {
            "script": "user.getAddresses() != null &&
user.getAddresses().size() > 0 ?
transformationUtil.getLookupCode(agcs_tenant_id, agcs_target_id, 'countries',
user.getAddresses().get(0).getCountry()) : null"
          },
          "uiProperties": {
            "inputType": "Auto",
            "widget": "SelectOne",
            "title": "Country",
            "labelHint": "Country",
            "minLength": 1,
            "maxLength": 256
          }
        },
        {
          "name": "managerLogin",
          "dataType": "TEXT",
          "usage": [
            "READ",
            "PROVISION"
          ],
          "uiProperties": {
```

```
      "inputType": "Admin",
      "widget": "Text",
      "title": "Manager Login",
      "labelHint": "Manager Login",
      "minLength": 1,
      "maxLength": 256
    }
  },
  {
    "name": "preferredLanguage",
    "dataType": "TEXT",
    "usage": [
      "READ",
      "PROVISION"
    ],
    "relationship": {
      "relatedTo": "languages",
      "relatedBy": "uid"
    },
    "uiProperties": {
      "inputType": "Admin",
      "widget": "SelectOne",
      "title": "Language",
      "labelHint": "Language",
      "minLength": 1,
      "maxLength": 256
    }
  },
  {
    "name": "userType",
    "dataType": "TEXT",
    "usage": [
      "READ"
    ]
  },
  {
    "name": "status",
    "dataType": "FLAG",
    "usage": [
      "READ",
      "PROVISION"
    ],
    "outboundTransformation": {
      "script": "true"
    },
    "uiProperties": {
      "inputType": "Auto",
      "widget": "Text",
      "title": "Status",
      "labelHint": "Status",
      "minLength": 1,
      "maxLength": 256
    }
  },
  {
    "name": "groups",
```

```
        "dataType": "TEXT",
        "nature": [
          "MULTIVALUED"
        ],
        "usage": [
          "READ",
          "PROVISION"
        ],
        "relationship": {
          "relatedTo": "GroupAsEntitlement",
          "relatedBy": "uid",
          "relationshipProperties": []
        },
        "uiProperties": {
          "inputType": "Admin",
          "widget": "RepeatableFieldSet",
          "title": "Groups",
          "labelHint": "Groups",
          "minLength": 1,
          "maxLength": 256
        }
      }
    ]
  },
  {
    "name": "countries",
    "type": "LOOKUP",
    "attributes": [
      {
        "name": "uid",
        "dataType": "TEXT",
        "nature": [
          "REQUIRED"
        ],
        "usage": [
          "READ"
        ]
      },
      {
        "name": "name",
        "dataType": "TEXT",
        "nature": [
          "REQUIRED"
        ],
        "usage": [
          "READ"
        ]
      }
    ]
  },
  {
    "name": "languages",
    "type": "LOOKUP",
    "attributes": [
      {
        "name": "uid",
```

```
            "dataType": "TEXT",
            "nature": [
              "REQUIRED"
            ],
            "usage": [
              "READ"
            ]
          },
          {
            "name": "name",
            "dataType": "TEXT",
            "nature": [
              "REQUIRED"
            ],
            "usage": [
              "READ"
            ]
          }
        ]
      }
    ]
}
```

Request Template Outline

In order to support requests to the required Authoritative Source or Managed System you need to create your request template using the supported outline.

Request Outline

The request outline you should follow when creating your request template is:

```
{
  "id": "", // request id
  "name": "", // request name
  "paginationType": "", // either of OFFSET,PAGE_INCREMENT,PAGE_TOKEN
  "method": "", // either of GET,HEAD,POST,PUT,PATCH,DELETE,OPTIONS,TRACE
  "uri": {
    "scheme": "", // request URI scheme either of http, https
    "host": "", // request URI host
    "path": "" // request URI path
  },
  "queryParameters": [
    {
      "name": "", // queryParameter name
      "value": "" // queryParameter value
    }
  ],
  "headers": [
    {
      "name": "", // request header name
      "value": "" // request header value
    }
  ],
  "body": {
        "type": "text",
```

```
        "textBody": {} // json request body
    },
    "subRequests": [ // subrequest if any
      {
        "id": "", // subrequest id
        "name": "", // subrequest name
        "paginationType": "", // either of OFFSET,PAGE_INCREMENT,PAGE_TOKEN
        "method": "", // either of GET,HEAD,POST,PUT,PATCH,DELETE,OPTIONS,TRACE
        "uri": {
          "scheme": "", // subrequest URI scheme either of http, https
          "host": "", // subrequest URI host
          "path": "" // subrequest URI path
        },
        "queryParameters": [
          {
            "name": "", // subrequestqueryParameter name
            "value": "" // subrequest queryParameter value
          }
        ],
        "headers": [
          {
            "name": "", // subrequest header name
            "value": "" // subrequest header value
          }
        ],
        "body": {
          "type": "text",
          "textBody": {} // json subrequest body
        }
      }
    ]
}
```

Pagination Support

The request outline supports a number of pagination methods to prevent large search results causing excessive network traffix. The following pagination methods are supported:

- OFFSET
- PAGE_INCREMENT
- PAGE_TOKEN

Parameters for these methods are detailed as follows. These should be added to the request outline.

Table 5-60 Pagination Support

Pagination Type	Prerequisite	Configuration
OFFSET	The REST API that you are integrating with Oracle Access Governance must support OFFSET pagination when returning a response.	If `paginationType` is set to OFFSET then you should add the following parameter values to the outline: `"queryParameters": [ {"name": "startIndex", "value": "<EL>currentOffset</EL>"} {"name": "count", "value": "<EL>limit</EL>"} ]`
PAGE_INCREMENT	The REST API that you are integrating with Oracle Access Governance must support PAGE_INCREMENT pagination when returning a response.	If `paginationType` is set to PAGE_INCREMENT then you should add the following parameter values to the outline: `"queryParameters": [ {"name": "currentPage", "value": "<EL>currentPage</EL>"} {"name": "pageSize", "value": "<EL>pageSize</EL>"} ]`
PAGE_TOKEN	The REST API that you are integrating with Oracle Access Governance must support PAGE_TOKEN pagination when returning a response.	If `paginationType` is set to PAGE_TOKEN then you should add the following parameter values to the outline: `"queryParameters": [ {"name": "top", "value": "<EL>pageSize</EL>"} {"name": "skiptoken", "value": "<EL>previousRequestResponseHeaders.get('token')==null ? '' : previousRequestResponseHeaders.get('token').get(0)</EL>"} ]`

> **Note:**
>
> The name for a particular pagination parameter may vary depending on the REST API that you are connecting with. For example, for OFFSET pagination, the parameters could be:
>
> ```
> {"name": "startIndex", "value": "<EL>currentOffset</EL>"}
> {"name": "count", "value": "<EL>limit</EL>"}
> ```
>
> or
>
> ```
> {"name": "beginIndex", "value": "<EL>currentOffset</EL>"}
> {"name": "increment", "value": "<EL>limit</EL>"}
> ```
>
> You should use the name specified by your API, but use the values as discussed in this article.

Request Template Output

Output of the request template is returned as a JSON document for defined entities and operations.

Request Template Output

Your request template output will look similar to that provided in the reference implementation (`<ReferenceBase>/functions/grc-schema-template/src/main/resources/request/applications/<YourApplicationName>/<EntityName>/<Operation>_TEMPLATE.json`):

For example, from the reference implementation:

- EntityName: UserAsIdentity
- Operation: GET

```
{
  "id": "1",
  "name": "Get User As Identity By ID",
  "method": "GET",
  "uri": {
    "scheme": "<<SCHEME>>",
    "host": "<<HOST>>",
    "path": "/admin/v1/Users/<EL>attributes.get('uid').get(0)</EL>"
  },
  "headers": [
    {
      "name": "Content-Type",
      "value": "application/json"
    },
    {
      "name": "Authorization",
      "value": "<<CREDENTIALS>>"
    }
  ]
}
```

- EntityName: UserAsIdentity
- Operation: SEARCH

```
{
  "id": "1",
  "name": "Search Users As Identity sort by displayName",
  "paginationType": "OFFSET",
  "method": "GET",
  "uri": {
    "scheme": "<<SCHEME>>",
    "host": "<<HOST>>",
    "path": "/admin/v1/Users"
  },
  "queryParameters": [
    {
      "name": "startIndex",
      "value": "<EL>currentOffset</EL>"
    },
    {
      "name": "count",
      "value": "<EL>limit</EL>"
    },
    {
      "name": "sortBy",
      "value": "displayName"
    }
  ],
  "headers": [
    {
      "name": "Content-Type",
      "value": "application/json"
    },
    {
      "name": "Authorization",
      "value": "<<CREDENTIALS>>"
    }
  ]
}
```

Response Template Outline

In order to support response format for identity and account data you need to create your response template using the supported outline.

Response Outline

The response outline you should follow when creating your response template is:

```
{
  "items": "", // items json path
  "attributes": [
    {
      "name": "", // attribute name
      "value": "" // attribute value json path
    },
```

```
    {
      "name": "", // attribute name for subrequest response
      "responseOfSubRequestId": "", // subrequest id
      "items": "", // items json path for subrequest response
      "subAttributes": [ // sub attributes for subrequest response
        {
          "name": "", // subrequest response attribute name
          "value": "" // subrequest response attribute value json path
        }
      ]
    }
  ]
}
```

Response Template Output

Output of the response template is returned as a JSON document for defined entities and operations.

Response Template Output

Your response template output will look similar to that provided in the reference implementation (`<ReferenceBase>/functions/grc-response-template/src/main/resources/response/applications/<YourApplicationName>/<EntityName>/<Operation>_TEMPLATE.json`):

For example, from the reference implementation:

- EntityName: UserAsIdentity
- Operation: GET

```
{
  "attributes": [
    {
      "name": "uid",
      "value": "<JP>$.id</JP>"
    },
    {
      "name": "name",
      "value": "<JP>$.userName</JP>"
    },
    {
      "name": "email",
      "value": "<JP>$.emails[?(@.primary == true)].value</JP>"
    },
    {
      "name": "firstName",
      "value": "<JP>$.name.familyName</JP>"
    },
    {
      "name": "lastName",
      "value": "<JP>$.name.givenName</JP>"
    },
    {
      "name": "displayName",
      "value": "<JP>$.displayName</JP>"
    },
```

```
    {
      "name": "usageLocation",
      "value": "<JP>$.addresses[?(@.type == 'work')].country</JP>"
    },
    {
      "name": "country",
      "value": "<JP>$.addresses[?(@.type == 'work')].country</JP>"
    },
    {
      "name": "managerLogin",
      "value": "<JP>$.
['urn:ietf:params:scim:schemas:extension:enterprise:2.0:User'].manager.value</
JP>"
    },
    {
      "name": "preferredLanguage",
      "value": "<JP>$.preferredLanguage</JP>"
    },
    {
      "name": "status",
      "value": "<JP>$.active</JP>"
    }
  ]
}
```

- EntityName: UserAsIdentity
- Operation: SEARCH

```
{
  "items": "<JP>$.Resources[*]</JP>",
  "attributes": [
    {
      "name": "uid",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].id</JP>"
    },
    {
      "name": "name",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].userName</JP>"
    },
    {
      "name": "email",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].emails[?(@.primary ==
true)].value</JP>"
    },
    {
      "name": "firstName",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].name.familyName</JP>"
    },
    {
      "name": "lastName",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].name.givenName</JP>"
    },
    {
      "name": "displayName",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].displayName</JP>"
```

```
    },
    {
      "name": "usageLocation",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].addresses[?(@.type ==
'work')].country</JP>"
    },
    {
      "name": "country",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].addresses[?(@.type ==
'work')].country</JP>"
    },
    {
      "name": "managerLogin",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].
['urn:ietf:params:scim:schemas:extension:enterprise:2.0:User'].manager.value</
JP>"
    },
    {
      "name": "preferredLanguage",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].preferredLanguage</JP>"
    },
    {
      "name": "status",
      "value": "<JP>$.Resources[<EL>currentIndex</EL>].active</JP>"
    }
  ]
}
```

Basic Authorization - Sample Token Creation Code

If you use basic authorization with your Generic REST orchestrated system you will need to create an authorization token. The sample code which follows, gives an example of how to code this function:

Sample Token Creation Code

If you use basic authorization with your Generic REST orchestrated system you will need to create an authorization token. The sample code which follows, gives an example of how to code this function:

```
public static String getAuthorizationValue(Config config) {
    String TOKEN_PREFIX_BASIC = "Basic ";
    try {
      String vaultJsonValue =
              VaultUtil.getDataFromVault(
                      config.getAuthenticationDetail().get("secretId"),
                      config.getAuthenticationDetail().get("region"));
      String username = VaultUtil.getAttributeValueFromJson(vaultJsonValue,
"username");
      String password = VaultUtil.getAttributeValueFromJson(vaultJsonValue,
"password");
      String authHeader = username.concat(":").concat(password);
      return TOKEN_PREFIX_BASIC
              + Base64.getEncoder()
              .encodeToString(authHeader.getBytes(StandardCharsets.UTF_8));
    } catch (UnsupportedEncodingException | JsonProcessingException e) {
```

```
      System.err.println("Exception occurred while getting secret from vault.
" + e.getMessage());
      throw new RuntimeException("Exception occurred while getting secret
from vault", e);
    }
  }
  // secretId and region value will come from config.yaml file
  // username and password value will come from vault secret which has been
configured in above steps.
```

OAuth Authorization - Sample Token Creation Code

If you use OAuth authorization with your Generic REST orchestrated system you will need to create an authorization token. The samples which follow, give an example of how to code this function:

Sample Token Creation Code

For IDCS orchestrated system:

This code is provided with the sample implementation that comes with the Generic REST Connector, and can be found at `<SampleBase>/grc-serverless-function-samples/idm-agcs-serverless-multi-application-sample/grc-commons/src/main/java/com/oracle/idm/agcs/grc/fn/commons/provider/IDCSAuthenticationProvider.java`.

```
/*
 * Copyright (c) 2024, Oracle and/or its affiliates. All rights reserved.
 * Licensed under the Universal Permissive License v 1.0 as shown at https://
oss.oracle.com/licenses/upl.
 */
package com.oracle.idm.agcs.grc.fn.commons.provider;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonNode;
import com.oracle.idm.agcs.grc.fn.commons.config.ConnectedSystemConfig;
import com.oracle.idm.agcs.grc.fn.commons.exception.ProcessingFailedException;
import com.oracle.idm.agcs.icfconnectors.commons.util.VaultUtil;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.InetSocketAddress;
import java.net.ProxySelector;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.nio.charset.StandardCharsets;
import java.text.MessageFormat;
import java.util.Base64;

public class IDCSAuthenticationProvider extends AuthenticationProvider {

  @Override
  public String getAuthorizationValue(ConnectedSystemConfig
connectedSystemConfig) {
    String url =
        connectedSystemConfig
```

```
            .getAuthenticationDetail()
            .get("scheme")
            .concat("://")
            .concat(connectedSystemConfig.getAuthenticationDetail().get("host"
))
            .concat(connectedSystemConfig.getAuthenticationDetail().get("path"
));
    String requestBody =
"grant_type=client_credentials&scope=urn%3Aopc%3Aidm%3A__myscopes__";
    String authHeader;
    try {
      String vaultJsonValue =
VaultUtil.getDataFromVault(connectedSystemConfig.getAuthenticationDetail().ge
t("secretId"), connectedSystemConfig.getAuthenticationDetail().get("region"));
      String clientCode =
VaultUtil.getAttributeValueFromJson(vaultJsonValue, "clientCode");
      String clientSecret =
VaultUtil.getAttributeValueFromJson(vaultJsonValue, "clientSecret");
      authHeader = clientCode.concat(":").concat(clientSecret);
    } catch (UnsupportedEncodingException | JsonProcessingException e) {
      System.err.println("Exception occurred while getting secret from vault.
" + e.getMessage());
      throw new ProcessingFailedException(
              "Exception occurred while getting secret from vault", e);
    }
    HttpClient client = HttpClient.newBuilder().build();
    String connectorProxyHost = System.getProperty(CONNECTOR_PROXY_HOST);
    String connectorProxyPort = System.getProperty(CONNECTOR_PROXY_PORT);
    if (null != connectorProxyHost
            && !connectorProxyHost.trim().isEmpty()
            && null != connectorProxyPort
            && !connectorProxyPort.trim().isEmpty()) {
      System.out.println(
              MessageFormat.format(
                      "connectorProxyHost {0} and connectorProxyPort {1} is
available in system property",
                      connectorProxyHost, connectorProxyPort));
      try {
        client =
                HttpClient.newBuilder()
                        .proxy(
                                ProxySelector.of(
                                        new InetSocketAddress(
                                                connectorProxyHost,
Integer.parseInt(connectorProxyPort))))
                        .build();
      } catch (NumberFormatException exception) {
        System.err.println("connectorProxyPort value is not integer :
"+exception);
      }
    }
    final HttpRequest request =
        HttpRequest.newBuilder()
            .uri(URI.create(url))
            .header(HEADER_NAME_CONTENT_TYPE,
HEADER_VALUE_CONTENT_TYPE_FORM_URL_ENCODED)
```

```
            .header(
                HEADER_NAME_AUTHORIZATION,
                TOKEN_PREFIX_BASIC
                    + Base64.getEncoder()
                        .encodeToString(authHeader.getBytes(StandardCharsets.U
TF_8)))
            .POST(HttpRequest.BodyPublishers.ofString(requestBody))
            .build();
    HttpResponse<String> response;
    try {
      response = client.send(request, HttpResponse.BodyHandlers.ofString());
    } catch (IOException e) {
      System.err.println(
          "IDCSAuthenticationProvider Auth Token API HttpRequest failed due
to IOException."
              + e.getMessage());
      throw new ProcessingFailedException(
          "IDCSAuthenticationProvider Auth Token API HttpRequest failed due
to IOException.", e);
    } catch (InterruptedException e) {
      System.err.println(
          "IDCSAuthenticationProvider Auth Token API HttpRequest failed due
to InterruptedException."
              + e.getMessage());
      throw new ProcessingFailedException(
          "IDCSAuthenticationProvider Auth Token API HttpRequest failed due
to InterruptedException.",
          e);
    }
    System.err.println(
        "IDCSAuthenticationProvider Auth Token API HttpResponse is :: " +
response.body());
    JsonNode jsonNode;
    try {
      jsonNode = objectMapper.readTree(response.body());
    } catch (JsonProcessingException e) {
      System.err.println(
          "IDCSAuthenticationProvider Auth Token API HttpResponse parsing to
json is failed.");
      throw new ProcessingFailedException(
          "IDCSAuthenticationProvider Auth Token API HttpResponse parsing to
json is failed.", e);
    }
    return
TOKEN_PREFIX_BEARER.concat(jsonNode.get(ACCESS_TOKEN_ATTRIBUTE).textValue());
  }
}
```

For AzureAD orchestrated system:

This code is provided with the sample implementation that comes with the Generic REST Connector, and can be found at `<SampleBase>/grc-serverless-function-samples/idm-`

agcs-serverless-multi-application-sample/grc-commons/src/main/java/com/oracle/idm/agcs/grc/fn/commons/provider/AzureAdAuthenticationProvider.java.

```
/*
 * Copyright (c) 2024, Oracle and/or its affiliates. All rights reserved.
 * Licensed under the Universal Permissive License v 1.0 as shown at https://oss.oracle.com/licenses/upl.
 */
package com.oracle.idm.agcs.grc.fn.commons.provider;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonNode;
import com.oracle.idm.agcs.grc.fn.commons.config.ConnectedSystemConfig;
import com.oracle.idm.agcs.grc.fn.commons.exception.ProcessingFailedException;
import com.oracle.idm.agcs.icfconnectors.commons.util.VaultUtil;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.InetSocketAddress;
import java.net.ProxySelector;
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;

public class AzureAdAuthenticationProvider extends AuthenticationProvider {
  @Override
  public String getAuthorizationValue(ConnectedSystemConfig connectedSystemConfig) {
    String url =
        connectedSystemConfig
            .getAuthenticationDetail()
            .get("scheme")
            .concat("://")
            .concat(connectedSystemConfig.getAuthenticationDetail().get("host"))
            .concat(connectedSystemConfig.getAuthenticationDetail().get("path"));
    String requestBody;
    try {
      String vaultJsonValue = VaultUtil.getDataFromVault(connectedSystemConfig.getAuthenticationDetail().get("secretId"), connectedSystemConfig.getAuthenticationDetail().get("region"));
      requestBody =
              "client_id="
                      + VaultUtil.getAttributeValueFromJson(vaultJsonValue, "clientCode")
                      + "&client_secret="
                      + VaultUtil.getAttributeValueFromJson(vaultJsonValue, "clientSecret")
                      + "&grant_type="
                      + connectedSystemConfig.getAuthenticationDetail().get("grantType")
                      + "&scope="
                      + connectedSystemConfig.getAuthenticationDetail().get("scope");
    } catch (UnsupportedEncodingException | JsonProcessingException e) {
```

```
      System.err.println("Exception occurred while getting secret from vault.
" + e.getMessage());
      throw new ProcessingFailedException(
              "Exception occurred while getting secret from vault", e);
    }
    final HttpClient client = getHttpClient(connectedSystemConfig);
    final HttpRequest request =
        HttpRequest.newBuilder()
            .uri(URI.create(url))
            .header(AuthenticationProvider.HEADER_NAME_CONTENT_TYPE,
AuthenticationProvider.HEADER_VALUE_CONTENT_TYPE_FORM_URL_ENCODED)
            .POST(HttpRequest.BodyPublishers.ofString(requestBody))
            .build();
    HttpResponse<String> response;
    try {
      response = client.send(request, HttpResponse.BodyHandlers.ofString());
    } catch (IOException e) {
      System.err.println(
          "AzureAdAuthenticationProvider Auth Token API HttpRequest failed
due to IOException."
              + e.getMessage());
      throw new ProcessingFailedException(
          "AzureAdAuthenticationProvider Auth Token API HttpRequest failed
due to IOException.", e);
    } catch (InterruptedException e) {
      System.err.println(
          "AzureAdAuthenticationProvider Auth Token API HttpRequest failed
due to InterruptedException."
              + e.getMessage());
      throw new ProcessingFailedException(
          "AzureAdAuthenticationProvider Auth Token API HttpRequest failed
due to InterruptedException.",
          e);
    }
    System.err.println(
        "AzureAdAuthenticationProvider Auth Token API HttpResponse is :: " +
response.body());
    JsonNode jsonNode;
    try {
      jsonNode =
AuthenticationProvider.objectMapper.readTree(response.body());
    } catch (JsonProcessingException e) {
      System.err.println(
          "AzureAdAuthenticationProvider Auth Token API HttpResponse parsing
to json is failed.");
      throw new ProcessingFailedException(
          "AzureAdAuthenticationProvider Auth Token API HttpResponse parsing
to json is failed.",
          e);
    }
    return
AuthenticationProvider.TOKEN_PREFIX_BEARER.concat(jsonNode.get(AuthenticationP
rovider.ACCESS_TOKEN_ATTRIBUTE).textValue());
  }

  private HttpClient getHttpClient(ConnectedSystemConfig
```

```
connectedSystemConfig) {
    if (null ==
connectedSystemConfig.getAuthenticationDetail().get("proxyHost")
        || null ==
connectedSystemConfig.getAuthenticationDetail().get("proxyPort")) {
      return HttpClient.newHttpClient();
    }
    return HttpClient.newBuilder()
        .proxy(
            ProxySelector.of(
                new InetSocketAddress(

connectedSystemConfig.getAuthenticationDetail().get("proxyHost"),

Integer.parseInt(connectedSystemConfig.getAuthenticationDetail().get("proxyPor
t")))))
        .build();
  }
}
```

Arcon Privileged Access Management (Arcon PAM)

Overview: Integrate Oracle Access Governance with

Oracle Access Governance can be integrated with , enabling identity orchestration, including on-boarding of identity (user) data, and provisioning of accounts.

can be integrated with Oracle Access Governance to ensure synchronized lifecycle management of privileged accounts within your enterprise, aligning with other identity-aware applications. offers identity management for various models, including Cloud Identity, Synchronized Identity, and Federated Identity, making it a valuable choice for organizations seeking consistent management of accounts, groups, and roles.

Integration Architecture Overview

You can perform full data load for accounts in . Once a connection is established, you can perform remediation tasks for user accounts, groups and roles.

Oracle Access Governance uses HTTPS to communicate with the API, which provides programmatic access through SCIM API endpoints. These endpoints enable Oracle Access Governance to perform create, read, and update operations on various directory data and objects, including users, roles, multi-factor authentication, services, and groups.

Functional Overview: Use Cases Supported for Integration

integration supports management of accounts from Oracle Access Governance, including the following use cases.

- Configure Orchestrated System

  See Configure Integration Between Oracle Access Governance and ARCON PAM.

- Match Identity and Account Attributes using Correlation Rules

Review or configure matching rules to match the identity and account data and build a composite identity profile. To view the default matching rule for this orchestrated system, see Default Matching Rules.

- Load Data

  Ingest accounts and roles that can be managed by Oracle Access Governance.

- Create Account

  Ingest account data from your orchestrated system or request an access for an identity. This allows you to provision entitlements (Role, Group, Service) and account details (Line of Business, Multi-factor Authentication).

- Update Account

  Update account details by assigning or removing permissions.This allows you to update entitlements (Role, Group, Service) and account details (Line of Business, Multi-factor Authentication).

- Enable/disable Account

  Enable or disable an account associated with an identity. This will either remove or restore accesses for the account.

Oracle Access Governance enables **API-based** seamless integration with enabling identity orchestration, automatic onboarding of accounts and roles, and reconciliation of accounts. Oracle Access Governance supports management of accounts as a **Managed System**.

Prerequisites

Before you install and configure an Orchestrated System, you should consider the following prerequisites and tasks.

1. Your system is certified with Oracle Access Governance. Refer to ARCON PAM Components Certified for Integration with Oracle Access Governance for details of the versions supported.

Configure

You can establish a connection between and Oracle Access Governance by entering connection details. To achieve this, use the orchestrated systems functionality available in the Oracle Access Governance Console.

Navigate to the Orchestrated Systems Page

The Orchestrated Systems page of the Oracle Access Governance Console is where you start configuration of your orchestrated system.

Navigate to the Orchestrated Systems page of the Oracle Access Governance Console, by following these steps:

1. From the Oracle Access Governance navigation menu icon  , select **Service Administration → Orchestrated Systems**.
2. Click the **Add an orchestrated system** button to start the workflow.

Select system

On the **Select system** step of the workflow, you can specify which type of system you would like to integrate with Oracle Access Governance.

You can search for the required system by name using the **Search** field.

1. Select .
2. Click **Next**.

Add details

Add details such as name, description, and configuration mode.

Add Owners

Add primary and additional owners to your orchestrated system to allow them to manage resources.

You can associate resource ownership by adding primary and additional owners. This drives self-service as these owners can then manage (read, update or delete) the resources that they own. By default, the resource creator is designated as the resource owner. You can assign one primary owner and up to 20 additional owners for the resources.

> **Note:**
>
> When setting up the first Orchestrated System for your service instance, you can assign owners only after you enable the identities from the Manage Identities section.

To add owners:

1. Select an Oracle Access Governance active user as the primary owner in the **Who is the primary owner?** field.
2. Select one or more additional owners in the **Who else owns it?** list. You can add up to 20 additional owners for the resource.

You can view the **Primary Owner** in the list. All the owners can view and manage the resources that they own.

Account settings

Outline details of how to manage account settings when setting up your orchestrated system including notification settings, and default actions when an identity moves or leaves your organization.

On the **Account settings** step of the workflow, enter details of how you would like to manage accounts with Oracle Access Governance when configured as a managed system:

1. Select where to send notification emails when an account is created. The default setting is **User**. You can select one, both, or none of these options. If you select no options then notifications will not be sent when an account is created.
   - User
   - User manager

2. When an identity moves within your enterprise, for example when moving from one department to another, you may need to adjust what accounts the identity has access to. In some cases the identity will no longer require certain accounts which are not relevant to their new role in the enterprise. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete
3. When an identity leaves your enterprise you should remove access to their accounts. You can select what to do with the account when this happens. Select one of the following options:
   - Disable
   - Delete

> **Note:**
>
> If you do not configure your system as a managed system then this step in the workflow will display but is not enabled. In this case you proceed directly to the **Integration settings** step of the workflow.

> **Note:**
>
> If your orchestrated system requires dynamic schema discovery, as with the Generic REST and Database Application Tables (Oracle) integrations, then only the notification email destination can be set (User, Usermanager) when creating the orchestrated system. You cannot set the disable/delete rules for movers and leavers. To do this you need to create the orchestrated system, and then update the account settings as described in Configure Orchestrated System Account Settings.

Integration settings

Enter details of the connection to your system.

1. On the **Integration settings** step of the workflow, enter the details required to allow Oracle Access Governance to connect to your system.

Table 5-61 Integration settings

Parameter Name	Mandatory?	Description
What is the host?	Yes	instance URL.
What is the port number?	No	Port number for the instance.
What is the authentication server URL to validate the client ID and client secret?	Yes	URL of the authentication server that validates the client ID and client secret for the target system(e.g :`https://[host]/auth/api/token/getToken`).
What is the username for authentication?	Yes	User name to connect to the instance.
What is the password/Confirm password	Yes	The password that authenticates the user you are connecting to the instance.

2. Click **Add** to create the orchestrated system.

Finish Up

Finish up configuration of your orchestrated system by providing details of whether to perform further customization, or activate and run a data load.

The final step of the workflow is **Finish Up**.

You are given a choice whether to further configure your orchestrated system before running a data load, or accept the default configuration and initiate a data load. Select one from:

- **Customize before enabling the system for data loads**
- **Activate and prepare the data load with the provided defaults**

Post Configuration

There are no post configuration steps associated with the system.

Components Certified for Integration with Oracle Access Governance

The components that you can integrate with are listed below.

Table 5-62 Certified Components

Component Type	Component
System	ARCON Privileged Access Management

Supported Configuration Modes for Integrations

Oracle Access Governance integrations can be setup in different configuration modes depending on your requirement for on-boarding identity data, and provisioning accounts.

Orchestrated System supports the following mode:

- **Managed System**
  You can manage accounts, groups and roles.

Supported Operations When Provisioning To

When you provision an account from Oracle Access Governance to certain operations are supported.

The Orchestrated System supports the following account operations when provisioning a user:

- Create User
- Update User
- Enable User
- Disable User
- Add Role
- Remove Role

- Add Group
- Remove Group
- Add Line of Business
- Remove Line of Business
- Add Multi-factor Authentication
- Remove Multi-factor Authentication
- Add Service
- Remove Service

> **Note:**
>
> Only Permanent services are currently supported. One-time and Time-based services are not currently supported.

For more details see Oracle Access Governance Integration Functional Overview and Integrate with ARCON PAM.

Default Supported Attributes

Oracle Access Governance supports the following default attributes.

These attributes are mapped depending on the direction of the connection, for example:

- Data being provisioned into from Oracle Access Governance:

  `account.lastName` will map to `User.name.familyName`

Table 5-63 Default Attributes for

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name
User	id	uid	Unique Id
	userName	name	User login
	displayName	displayName	Name
	ValidTillDate	endDate	End date
	emails.value	emails	Email
	domainName	domainName	Domain name
	phoneNumbers.value	phone	Phone
	userTypeId	userType	User type
	name.formatted	fullName	Full name
	name.familyName	lastName	Last name
	name.givenName	firstName	First name
	name.middleName	middleName	Middle name
	LobPrimary	primaryLob	Primary line of businesses
	IsActive	status	Status
	password	password	Password
Roles		roles	Roles

Table 5-63 (Cont.) Default Attributes for

Entity	Account Attribute	Oracle Access Governance Account Attribute	Oracle Access Governance Identity attribute display name
Groups		groups	Groups
LOBS		lobs	Line of businesses
Multi-factor Authentication	userDualAuthFactType	mfas	Multi-factor authentications
Services	AccessTypeId	accessTypeId	Access type
	StartDateTime	accessDurationStartDate	Access duration start date
	EndDateTime	accessDurationEndDate	Access duration end date
	hours	perSessionHours	Per session duration in hours
	minutes	perSessionMinutes	Per session duration in minutes
	StartTime	accessPeriodStartTime	Access period start time
	EndTime	accessPeriodEndTime	Access period end time

Default Matching Rules

In order to map accounts to identities in Oracle Access Governance you need to have a matching rule for each orchestrated system.

The default matching rule for the orchestrated system is:

Table 5-64 Default Matching Rules

Mode	Default Matching Rule

Managed System
Account matching checks if incoming accounts match with existing identities.
	Screen value:
	`User login = Employee user name`
	Attribute name:
	`Account.name = Identity.userName`

6

Who Has Access to What

Who Has Access to What: Comprehensive Access Profile Visibility across Enterprises

Get comprehensive visibility on all the components, access information, and resources within an enterprise framework using Oracle Access Governance **Who has Access to What → Enterprise-wide Browser**. With this, you can track and monitor who has access to different systems, data, and applications, at what permission level, and for what it is being used, to make informed decisions and detect potential security issues for effective governance.

With **Enterprise-wide Browser**, you can get details into access profiles of identities across enterprise. As a manager, you can go to **My Directs' Access** to view access profiles of your team, enabling you to oversee the access privileges of team members. As an Oracle Access Governance user, you can view your own accesses from the **My Stuff**, and then **My Access** page. For more information on what access components you can view, refer to Enterprise-wide Access Profile Reference

- **Applicable Roles for Enterprise-wide Browser**: *Enterprise-wide Access Administrator* or *Administrator*
- **Applicable Roles for My Directs' Access and My Access**: *User*

Why Enterprise-wide Access Visibility Matters?

Enterprises today use multiple systems and go through various personnel changes in their ecosystem. With the extensive use of digital platforms, cloud services, physical assets, and interconnected systems, it is vital to safeguard information. Enterprises need to have a centralized view of all identity and access management information to identify vulnerabilities, detect risk, foresee potential gaps, and adjust policies to better protect the security posture of their organization.

Oracle Access Governance **Who has Access to What - Enterprise-wide Browser** enables enterprises to get access profile details into an enterprise infrastructure, providing a unified view of the security posture. You can get thorough insights into identities, resources, permissions, access bundles, accounts, policies, roles, group membership, organizations, and so on.

Use advanced filters to narrow down your search and locate specific infrastructure components within the enterprise. For example, you can locate information based on identity location, job code, department, or any other specific identity attribute. You can personalize the access insights dashboard by customizing (hiding/showing) the available information for a better user experience. By gaining deep visibility into enterprise security information, you can help enforce proper compliance and proactively validate that appropriate access has been granted throughout your enterprise.

Browsing Views: Selection Parameters to Explore Access Profile across Enterprise

You can view access information using various perspectives, such as Identities, Identity Collections, Roles, Permissions, Policies, Resources, and Organizations. Each one offers a different angle to view access information across the enterprise.

You can perform search or apply filters to refine your search results. For easier navigation, you can use quick links to traverse through and back various detail pages while exploring access insights. Using quick links, you can move up to five (5) levels, with the first one being as the **Enterprise-wide Browser** page, and the remaining four (4) are the four (4) most recently visited pages.

Refresh Timings

For entities local to Oracle Access Governance, you can immediately view them on the **Enterprise-wide Browser** page; the ingested access information is refreshed and visible as per the orchestrated system data load schedule. On the details page, reference counts are updated every hour to provide you with the latest counts. If you notice an entity listed before the count is updated, this is likely due to scheduled refresh timings followed by the service.

Here's what each browsing component offers:

Identities

Get a comprehensive view of all the available **Active** identities in Oracle Access Governance. You can view identity details displaying identity attributes along with intelligent risk insights generated by Oracle Access Governance. For example, you can view the access profile information of an intern working in your enterprise to ensure they are not part of any strategic group or have access to any critical resource. Browse an individual identity to get access details for the selected identity.

You will see the same set of information when exploring your own accesses within Oracle Access Governance, or when managers view identity details for their team members. For more information on what components you can view, refer to Enterprise-wide Access Profile Reference.

Identity Collections

Get comprehensive access insights at a group-level using **Identity Collections** as your perspective view. You can view identity collections created within Oracle Access Governance or groups in Oracle Cloud Infrastructure (OCI). It displays the basic details, such as status, number of members, or owner of the identity collection. This view is beneficial if you want to view access profiles for a group, rather than an individual. For example, you may want to ensure that only the identities part of the Finance department have access to critical accounting systems and data.

For OCI groups, the **Owner** is marked as **AG System**. Browse an individual identity collection to view access summary, reference count summary, its creation rule, intelligent insights, and member details. Additionally, you can run access reviews and download a PDF report presenting Identity collection details. For more information on what components you can view, refer to Enterprise-wide Access Profile Reference.

Organizations

Get a list of organizations created within Oracle Access Governance, displaying basic details, such as name, status, member count, and owner details. Browse an individual organization to

view its membership rules and member details. These are beneficial for looking out for hierarchical access within an enterprise. You can easily set up rules based on your department when creating an Organization. For example, in a banking system, regional branch managers can get access profile information of all the employees across all the branches managed by them. For more information on what components you can view, refer to Enterprise-wide Access Profile Reference.

Permissions

Get a list of all the permission types, such roles, privileges, groups, access bundles, and permissions for each resource ingested from an Orchestrated system. Browse an individual permission to see a reference count summary related to the selected permission along with the included resource access. For example, you may want to get a list all the available permissions associated with a critical resource in a tenancy. Within Permissions perspective, the viewable details for each permission varies based on **Granted permission type** and **Access Governance type**. For example, for **Access Governance type Access Bundle**, you can view associated identities, identity collections, roles, and policies but for **Access Governance type Roles**, you can view identities.

For more information on what components you can view, refer to Enterprise-wide Access Profile Reference.

Policies

Get a list of all the Oracle Cloud Infrastructure (OCI) policies and Oracle Access Governance policies, displaying basic policy details. Browse an individual policy to see a reference count summary related to the selected policy along with policy details, such as policy statements or associated roles or access bundles. For example, you can view all the available OCI policies for a specific compartment. Another example can be if you want to view what resources are assigned to an Identity Collection and its association for a specific Oracle Access Governance policy.

If you browse through an OCI policy, you will see OCI policy statements, Identities, Identity Collections (OCI groups), Resources. If you browse through Oracle Access Governance policy, you will see Identities, Identity Collections, Resources, Access Bundles, and Roles. For more information on what components you can view, refer to Enterprise-wide Access Profile Reference.

Resources

Get a list of enterprise resources and resource types across various systems or cloud tenancies integrated with Oracle Access Governance. You can also fetch which identities are currently assigned to that resource. For example, you may want to get access details related to a specific OCI bucket.

Manage a large set of resources by applying sorting techniques. Use the **Sort by** drop-down to sort resources by **Resource Name** or **Resource Type** and/or use the **Sort Direction** drop-down to arrange resources alphabetically either in ascending (A-Z) or descending (Z-A) order. Browse an individual resource to view access summary, identity details, or allocation insights based on *Organization*, *Source Organization* **(Source Org)**, *Job Code*, or *Location*. For more information on what components you can view, refer to Enterprise-wide Access Profile Reference.

Roles

Get list of the roles created in Oracle Access Governance. These roles are associated with identities via policies or self-service requests so that appropriate accesses can be assigned to identities to perform a task. For example, you can create a database administrator role in

Oracle Access Governance to associate database applications and privileges to identities or identity collections. For more information on what components you can view, refer to Enterprise-wide Access Profile Reference Roles Reference.

Search Capabilities: Using Keywords, Suggested and Advanced Filters

You can use our search capabilities to get specific and relevant results. You can use a basic keyword search for anything that you want to locate within an Enterprise-wide Browser, scope your search using the suggested filters, or apply advanced filters to further specify your search criteria and improve results.

Basic Keyword Search

The basic search covers keyword search, where you can input terms directly related to your requirement. For example, typing “IND” in the search box returns all the available rows that contain this match. This search can locate beyond the default view visible to you, and can find across all the available columns, including the hidden columns from the view.

Suggested Filters

You can scope the search and enhance the results by using the available suggested filters. These filters will vary based on the browsing selection you opted. For example, for Identities, you can limit your search to view workforce identities only by applying the **Access Governance subtype Workforce** filter.

Advanced Filters

If you are looking for something specific, then use Advanced filters. The attributes and logical operators vary based on the perspective view selected. For example, to view specific policy details in a compartment, you can use advanced filters to choose the compartment name. Another example would be if you want to view access information for the permissions assigned after a certain date, then you would use the **Created Date** advanced filter to specify the date. These are helpful to perform advanced searches to get specific result quickly.

Let’s see how you can use all three search capabilities to get improved results. For example, you want to locate the Workforce identity containing the keyword *Alison*, belonging either to *India* or *Chile* location. To search this,

- Type “Alison” in the search box for your basic search.
- Apply suggested filters on the **Access Governance subtype Workforce** filter.
- Add an advanced filter on Location, selecting *India* or *Chile*.

Usage Examples: Monitoring Access Profile Details in an Enterprise

Let's look at a few scenarios to understand how you can use the **Who has Access to What** feature to its full potential.

Scenario 1: Monitoring Access Profile Insights for the High-Risk Interns

You want to review the access profile of interns, having job codes IN101 and IN102, to get comprehensive insights of accesses assigned to them.

To do this:

1. Log on to Oracle Access Governance as an *Enterprise-wide Access Administrator*
2. Navigate to **Who has Access to What**, and then **Enterprise-wide Browser**.

3. In **Select what you want to browse list**, select **Identities**.
4. Apply the suggested filter on **Insights High Risk**.
5. Add an advanced filter on **Job code** *Equals* **IN101** or **IN102**.
   You will get a list of high-risk identities fulfilling the applied conditions.
6. For an identity, click the **View Details** button to view its assigned permissions, roles, policies, organizations, identity collection, resources, and accounts.
   You can further generate user-created access reviews for such high-risk identities so that reviewers can approve or revoke accesses appropriately.

> **Note:**
>
> Alternatively, you can create an identity collection for Interns by applying the job code membership rule. You can browse this identity collection from the **Enterprise-wide Browser** page. You can generate access review campaigns or provision the identities using this identity collection.

Scenario 2: Exploring OCI Policies across an OCI tenancy

As a cloud administrator, you want to view all the available Oracle Cloud Infrastructure (OCI) policies across a specified tenancy (*ewbgov*) to get a clear overview of access controls, permissions and resources allocated.

To do this:

1. Log on to Oracle Access Governance as an *Enterprise-wide Access Administrator*
2. Navigate to **Who has Access to What**, and then **Enterprise-wide Browser**.
3. In the **Select what you want to browse list** list, select **Policies**
4. Apply suggested filter on **Provider Oracle Cloud Infrastructure**.
5. Add an advanced filter on **Cloud account name** *Equals* **ewbgov**.
   The layout displays the count and list of all the OCI policies fulfilling the applied conditions.
6. For a policy, click the **View Details** button to view policy statements, identities, identity collections (OCI groups), or resources associated with this policy.

Access Reviews in Enterprise-wide Browser

You can run access reviews for identities, identity collections, policies from the Enterprise-wide Browser dashboard. This is also called "User-Created Access Reviews." You can directly initiate Identity access reviews or Access control reviews while exploring the access profile insights. For an identity, you can create access reviews to verify granted roles, access bundles, accounts, or permissions.

Scope: Different from generating periodic or ad-hoc campaigns which are part of planned access review audits in an enterprise. Generally, you need user-created access reviews to generate spontaneous access reviews. For example, you may generate access reviews for an identity where you identify access anomalies while exploring access profile for that identity. The reviewer can then validate these access reviews from the **My Access Reviews** page following the process defined in Perform Access Reviews. On the **My Access Reviews** page, you can differentiate these access reviews from the ones generated by campaigns by viewing the **Review Source** column. The ones marked with **User created** are initiated from the **Enterprise-wide Browser**. You can generate the following user-created access reviews:

- **Identity Access Reviews**: When you want to review accesses granted to an identity and verify that the assigned access rights are appropriate. For example, you can generate user-created access reviews to review accounts, permissions, and roles assigned to a high-risk identity.
  - If you select **Actions**, and then **Create access review** from the top banner of the details page, then all the possible review tasks for that identity including role, permission, and account will be generated for that identity.

  - If you select , and then **Create access review**, then access review for that specific permission, role, or account will be created. Refer to the Access Review Creation Criteria section to know for what permissions you can generate access reviews.
- **Access Control Reviews**: When you want to review the implementation of access controls, such as performing review on Identity Collection assigned to an identity, or reviewing or auditing accesses assigned through the Oracle Access Governance or Identity and Access Management (IAM) policy.

Access Review Creation Criteria

You can generate individual access reviews for a specific permission or role meeting the criteria given in the following table. For example, you cannot review permissions granted through a policy. You can review directly assigned permissions (`DIRECT`) or permissions granted within Oracle Access Governance. Here's the criteria for generating user created access reviews.

Table 6-1 Criteria to run Access Reviews from Enterprise-wide Browser

System	Valid Entities to Review	Valid Conditions for Review
Oracle Identity Governance (OIG)	• Accounts • Permissions • Identities • Roles	**Grant Type for Permission** • Request • Direct Provision • Reconciliation • Bulk Load **Account Status** • Provisioned • Active **Grant Type for Role** • Direct • Request
OCI	• Roles • Identity Collections or OCI Groups • Identities • Access Bundles	**Grant Type for Role Membership** • SERVICE_MANAGER_TO_USER • ADMINISTRATOR_TO_USER **Grant Type of Permission Managed by Oracle Access Governance** • Request

Table 6-1 (Cont.) Criteria to run Access Reviews from Enterprise-wide Browser

System	Valid Entities to Review	Valid Conditions for Review
Oracle Access Governance	• Identities • Accounts • Policies • Identity Collections • Roles • Access Bundles • Permissions	**Grant Type for Permissions** • Request • DIRECT **Account Status** • Provisioned • Active

User-Created Access Review Report

Generate a monthly report on access reviews created from **Enterprise-wide Browser** by selecting the **View user created access review report** button. You can generate a report based on the date range and access review type.
Similar to Campaign, you see a report displaying access review details and a breakdown of pending, approved, or revoked access review decisions for user role, user account and permission.

- For identity access reviews, you see the bifurcation of the review decisions based on top five organization, source organization, roles, and applications.
- For access control reviews, you see a breakdown of pending, approved, revoked, or modified access review decisions along with grouping of top five created since date ranges for identity collections or policies.

In addition to viewing report, you can also save the reports offline in PDF format or download the CSV data for record-keeping or further analysis or audit.

Explore Access Profile in an Enterprise

As an *Enterprise-wide Access Administrator*, you can use the **Who has Access to What** functionality of Oracle Access Governance to get 360-degree visibility of access maps across an enterprise. Here, you get a complete and comprehensive understanding of identity access profiles across an enterprise. Managers can check accesses for their direct reports.

View Access Profile across Enterprise

As an *Enterprise-wide Access Administrator*, get visibility into accesses across an enterprise using the **Enterprise-wide Browser** functionality. There are multiple perspective views available to browse through the access profile information. You can browse to view identity details, their accesses including roles, resources, permissions, accounts, or policies. Alternatively, you can choose access control entities to browse access information.

Here's how you can view access profile for your enterprise:

1. Log on to Oracle Access Governance as an Enterprise-wide Access Administrator.
2. From the  navigation menu, select **Who has Access to What**, and then **Enterprise-wide Browser**.
3. In the **Enterprise-wide Browser** page, select the perspective view in the **Select what you want to browse** list.
4. Perform keyword search or select suggested filters for anything to you want to locate.

5. To search something specific, use the **Add an advanced filter** button to apply advanced filters.

   Based on the perspective view selected, you will see the access information in a grid view. You can customize this default view by hiding/showing the columns.

6. To view details related to each record in a grid view, select the **View details** link.

   In the details page, you will see the reference count summary along with associated information for that perspective. You can further explore access details or identity information by selecting the **View details** link throughout the view. For more information on what components you can view, refer to Enterprise wide Access Profile Reference .

Search and Apply Filters

You can retrieve data of your choice by performing keyword search or applying filters. Within **Enterprise-wide Browser** page, you can apply suggested filters to view focused information, or add an advanced filter to get some specific information.

Here's how you can apply search

1. To perform keyword search, in the **Enterprise-wide Browser** page, in the *Search* field, enter a keyword of your choice and press the **Enter** key on your keyword. Repeat the process to add additional keywords.

   Records containing the search phrase will be displayed. For more than one keywords, all the keywords need to be satisfied.

2. To view focused results, select one or more suggested filters available directly under the *Search* field.

3. To look out for something specific, add one or more advanced filters:

   a. Click the **Add an advanced filter** button.

   b. Select one of the identity attributes in the **What do you want to filter on?** list.

   c. Select the logical operator of your choice.

   d. Select the value and click **Apply**.

   You can view the applied advanced filters under the **View advanced filters** collapsible section.

4. In the **View advanced filter** section

   a. Select the  edit icon to modify the value listed in the applied filter.

   b. Select the  delete icon to remove an advanced filter

   c. Click the **Clear all** link to remove all the applied advanced filters at once.

Manage Access Profile Layout

You can customize the default access profile layout by including (show) or excluding (hide) the data columns, sorting individual column, or reordering the data columns to change the view sequence. You can even download the CSV file for the first 500 records for offline analysis.

Hide or Show Columns in the Access Profile Layout

1. In the **Enterprise-wide Browser** page, select the  **Edit list setting** icon.
2. To add columns in your layout, from the **Hide** section, select one or more fields. The selected columns will be part of the **Show** section.
3. To remove columns from your layout, from the **Show** section clear one or more fields. These unselected columns will be part of the **Hide** section and will be hidden from the access profile view layout.
4. Click **Apply**.
5. To restore your default layout, select the **Restore defaults** link.

Reorder Columns in the Access Profile Layout

1. In the **Enterprise-wide Browser** page, select the  **Edit list setting** icon.
2. To reorder and rearrange the columns, select and drag the  drag-handle icon and drop it at the relevant position.
3. Click **Apply**.

Download Reports

- In the **Enterprise-wide Browser** page, select the  CSV icon to download the first 500 records available in the access profile view.
- To download the PDF screenshot of the access details visible to you in the application:
  - In the **Enterprise-wide Browser** page, for a specific access record, select the **View details** link.
  - From the **Actions** menu, select **Download screenshot PDF**.

> **Note:**
>
> If you're having trouble viewing PDFs downloaded using Mozilla Firefox, update the browser advanced font settings, and select the **Allow pages to choose your own font, instead of your selections above** check box.

Generate User-Created Access Reviews

Enterprise-wide Access Administrator or Oracle Access Governance *Administrator* can generate identity and access control access reviews while exploring the access profile insights within **Enterprise-wide Browser**. You can generate access reviews for identities, policies, and identity collections. For an identity, apart from regular identity access reviews, you can request specific access review on permissions, accounts, roles, identity collections, and policies.

Identity Access Reviews

1. Log on to Oracle Access Governance as an *Enterprise-wide Access Administrator*.
2. From the navigation menu, select **Who has Access to What**, and then **Enterprise-wide Browser**
3. In the **Enterprise-wide Browser** page, select **Identities** as the perspective view in the **Select what you want to browse** list.
4. Select the **View details** link corresponding to the identity for which you want to generate the review.
5. To create:
   - All the possible identity access reviews for that identity, from the **Actions** menu, select **Create access review**.
   - Specific identity access review for the selected permission, from the **More Actions** icon, select **Create access review**.
6. Select the reviewer in the **Who do you want to do the review** field.
7. Enter justification for this access review in the **Why are you wanting it reviewed?** box.
8. Click **Create**.

Access Control Access Reviews

1. Log on to Oracle Access Governance as an *Enterprise-wide Access Administrator*.
2. From the navigation menu, select **Who has Access to What**, and then **Enterprise-wide Browser**
3. In the **Enterprise-wide Browser** page, depending on the access control type, select either **Identity Collections** or **Policies** as the perspective view in the **Select what you want to browse** list.
4. Select the **View details** link corresponding to the access control that you want to review.
5. In the details page, from the **Actions** menu, select **Create access review**.
6. Select the reviewer in the **Who do you want to do the review** field. By default, the review is assigned to the owner of the selected access control.
7. Enter justification for this access review in the **Why are you wanting it reviewed?** box.
8. Click **Create**.

Reviewers can view these review tasks on the **My Access Reviews** page.

My Directs' Access - View Access Profile Information for Team

Managers can view comprehensive identity details for their direct reports. Managers can view what resources and accounts are assigned to each one, assigned oles at what permission level and with what groups they are associated.

> **Note:**
>
> This feature is not available if you have integrated only OCI IAM, as your target system, with Oracle Access Governance. Here, **My Directs' Access** will provide the access information only for the direct reports, and not skip-level reports.

To review team access:

1. In the Oracle Access Governance Console, select **Who Has Access to What**, and then **My Directs' Access** from the navigation menu.

   You navigate to the **My Directs' Access** page. A list of your direct reports is displayed. You can use **Advanced filters** to limit the search results of users specific to user attributes, such as an application, a job code, or a location.

2. To view access privileges assigned to a user, select the **Username**.

   You will see the identity details page displaying all the access components for the identity. Refer to Enterprise-wide Access Insights Reference.

Enterprise-wide Access Profile Reference

Use the **Enterprise-wide Browser** functionality of Oracle Access Governance to view access insights available across your enterprise. You can view access information using various perspectives, such as Identities, Identity Collections, Roles, Permissions, Policies, Resources, Organizations. Each one offers a different angle to the enterprise access information.

Get the reference information of details provided by each access component.

Identities

While exploring your access profile details, you can view your associated roles, permissions, accounts, ownership, organizations, identity collections, identity attributes, cloud resources, and policies.

For Identities, you can see the following information:

Table 6-2 Identity Access Profile Information

Access Component	Description
Identity Collections	Count and details of the identity collection associated with the identity. This can either be Oracle Access Governance identity collection or an ingested identity collection, such as OCI groups.

Table 6-2 (Cont.) Identity Access Profile Information

Access Component	Description
Permissions	Count and access rights detail associated with this identity. It gives clarity of how this access was granted, for which resource this permission has been granted, and whether it is a role, permission, or a privilege assigned to the identity.
Organizations	Count and details of Oracle Access Governance organizations associated with the selected identity.
Accounts	Get count and account details associated with this identity. It gives you details like account name, the orchestrated system name associated with the account, resource name, how the access has been granted, password change status. When viewing your own accesses using the **My Access** menu option, if the account is provisioned within Oracle Access Governance and **Password Change status** flag is set to **Applicable** , then you can change your password. To do so, select **Change password** and follow the instructions to change your password.
Roles	Count and details of roles assigned to this identity using the Oracle Access Governance Access Control framework. If you want to see the ingested roles available from Managed Systems, then see the **Permissions** tab.
Policies	Count and details of policies used for granting access to the selected identity. You can further browse a policy to view policy statement details by selecting the **View details** link. The policies assigned can either be Oracle Access Governance policies or cloud policies ingested from OCI.
Cloud Resources	Count and cloud resource details that specify resource name, its type, the associated privilege granted to the identity along with the policy name that granted this privilege.
Ownership	Count and details of access controls components owned by this identity, such as identity collections, roles, policies,
Identity Attributes	Core and custom Identity attributes along with its value. The attributes are logically sectioned under meaningful headings for relevancy.

Identity Collections

While exploring access insights across an enterprise, you can choose **Identity Collections** as your perspective view. This gives you the ability to view access management information at a group level. You can view identity collections created within Oracle Access Governance or Oracle Cloud Infrastructure (OCI) groups ingested into Oracle Access Governance.

From the **Identity Collections** perspective view, you can view the following details:

Table 6-3 Identity Collections Reference

Access Component	Description
Access Summary	For an identity collection, get access information, such as policies associated with this identity collection. • For Oracle Cloud Infrastructure (OCI) Identity Collections, you can view details related to policy statements associated with each policy. • For Oracle Access Governance Identity Collections, you can view policies and its association type like access bundles or role. Here, you can view the association details that lets you know the permissions and resources assigned to the selected identity collection. You can further navigate to view each component and its details.
Identities	List of identities or members part of the selected identity collection. Select the **View details** link to get access details on each identity. Click the  bar chart icon to view Member summary. Here, you can view the bifurcation of members in the identity collection based on source organization, organization, job code, location, or employee type.
Resources	Get count and access information on resources associated with this identity collection. From the reference count summary section, click the resource count to view resource details, such as its name, type and number of identities having access to this resource. You can browse further to view details of each resource.
Access Bundles	Get count and access information on access bundles associated with this identity collection. From the reference count summary section, click the Access bundles count to view its details, such as its name, granted permission type, its associated orchestrated system, resource and resource type. You can browse further to view details of each access bundle.
Policies	Get count and access information on policies associated with this identity collection. From the reference count summary section, click the Policies count to view its details, such as its name, provider (OCI or Oracle Access Governance). owner, and so on. You can browse further to view association and reference details of each policy.
Roles	Get count and access information on roles associated with this identity collection. From the reference count summary section, click the roles count to view its details, such as its name, who can request this role and its status. You can browse further to view access information and reference details of each role.

Organizations

While exploring access insights across an enterprise, you can choose **Organizations** as your perspective view. You can view organizations created within Oracle Access Governance.

From the **Organizations** perspective view, you can view the following details:

Table 6-4 Organizations Reference

Access Component	Description
Organization Details	For an organization, get its creation rules along with identity list part of this organization.
Identities	List of identities or members part of the selected organization. Select the **View details** link to get access details on each identity. Click the  bar chart icon to view Member summary. Here, you can view bifurcation of members in the organization based on source organization, organization, job code, location, or employee type.

Permissions

While exploring access insights across an enterprise, you can choose **Permissions** as your perspective view. This gives you the ability to view available permissions, roles, access bundles associated with various resources.

From the **Permissions** perspective view,**Granted permission type** and **Access Governance type** determines what entity details you can view for Permissions. For example, for **Access Governance type Access Bundle**, you can view associated identities, identity collections, roles, and policies.

Table 6-5 Permission Details

Access Governance Type	Granted Permission Type	Viewable Details
Access Bundle	Access Bundle	• Identities • Identity Collection • Roles • Policies

> **Note:** Applicable for Oracle Access Governance access bundles.

Table 6-5 (Cont.) Permission Details

Access Governance Type	Granted Permission Type	Viewable Details
Role	Role	Identities

Note:

Applicable for Oracle Cloud Infrastructure (OCI), and Oracle Identity Governance (OIG) roles. To view access details about Roles defined within Oracle Access Governanc

Table 6-5 (Cont.) Permission Details

Access Governance Type	Granted Permission Type	Viewable Details
	e, select **Roles** as your perspective view.	
Permission	Ingested permissions loaded from Managed systems and have not been provisioned using Oracle Access Governance, such as Permission, Privilege (applicable for Database systems), AD Group, and so on.	Identities

Table 6-6 Permissions Reference

Access Component	Description
Identities	Get count and list of identities who are assigned a specific role, entitlement or access bundle. From the reference count summary section, click the **Identities** to view a list of identities having access to the associated permission. Select the **View details** link to further browse each identity.
Policies	Get count and access information on policies to which this access bundle is associated. From the reference count summary section, click the **Policies** count to view its details, such as its name, provider Oracle Access Governance), owner, and so on. You can browse further to view association and reference details of each policy.
Identity Collections	Get count and access information on identity collections that has access to the selected access bundle. From the reference count summary section, click the **Identity Collections** count to view its details, such as its name, owner, member count, status and so on.
Roles	Get count and access information on roles to which this access bundle is associated. From the reference count summary section, click the roles count to view its details, such as its name, who can request this role and its status. You can browse further to view access information and reference details of each role.

Policies

While exploring access profile details for an enterprise, you can choose **Policies** as your perspective view. Here, you can browse through Oracle Access Governance policies and Oracle Cloud Infrastructure policies.

If you browse through an OCI policy, you will see OCI policy statements, Identities, Identity Collections (OCI groups), Resources, and if you browse through Oracle Access Governance policy, you will see Identities, Identity Collections, Resources, Access Bundles, and Roles.

Table 6-7 Policies Reference

Access Component	Description
Identities	List of identities assigned permissions and resources through this policy. From the reference count summary section, click the **Identities** to view a list of identities having access to the associated policy. Select the **View details** link to further browse each identity.
Identity Collections	List of identity collection assigned permissions and resources through this policy. From the reference count summary section, click the **Identity collections** to view a list of identity collections having access to the associated policy. Select the **View details** link to further browse each identity collection.
Access Bundles	Get count and access information on access bundles associated with this policy. This is applicable only for Oracle Access Governance policy. From the reference count summary section, click the **Access bundles** count to view its details, such as its name, granted permission type, its associated orchestrated system, resource and resource type. You can browse further to view details of each access bundle.
Resources	Get count of resources associated with this policy and resource details. From the reference count summary section, click the resource count to view resource details, such as its name, type and number of identities having access to this resource. You can browse further to view details of each resource.
Roles	Get count and access information on roles associated with this policy. You can view the identities association details and to what they have access. From the reference count summary section, click the roles count to view its details, such as its name, who can request this role and its status. You can browse further to view access information and reference details of each role.

Resources

While exploring access insights across an enterprise, you can choose **Resources** as your perspective view. This gives you the ability to view resource and its details ingested into Oracle Access Governance.

From the **Resources** perspective view, you can view the following details:

Table 6-8 Resources Reference

Access Component	Description
Identities	List of identities or members having access to the selected resource. From the reference count summary section, click the Identities count link to view a list of identities having access to the associated policy. Select the **View details** link to further browse each identity.
Identity Collections	Get count and access information on resources associated with this identity collection. From the reference count summary section, click the Resource count link to view resource details, such as its name, type and number of identities having access to this resource. You can browse further to view details of each resource.
Access Bundles	Get count and access information on access bundles that associate this resource for granting access to identities. From the reference count summary section, click the Access bundles count link to view its details, such as its name, granted permission type, its associated orchestrated system, resource and resource type. You can browse further to view details of each access bundle.
Policies	Get count and access information on policies granting access to this recource. From the reference count summary section, click the Policies count link to view its details, such as its name, provider (OCI or Oracle Access Governance). owner, and so on. You can browse further to view association and reference details of each policy.
Roles	Get count and access information on roles having access to this resource. From the reference count summary section, click the Roles count link to view its details, such as its name, who can request this role and its status. You can browse further to view access information and reference details of each role.
Accounts	Get count and access information on accounts having access to this resource. From the reference count summary section, click the Accounts count link to view its details.
Permissions	Get count and access information on permissions available to manage different level of access for this resource. From the reference count summary section, click the Permissions count link to view its details.

Roles

While exploring access insights across an enterprise, you can choose **Roles** as your perspective view. This gives you the ability to view roles created within Oracle Access Governance.

From the **Roles** perspective view, you can view reference count information along with the included access. Here are the following details:

Table 6-9 Roles Reference

Access Component	Description
Identities	List of identities or members who are assigned a specific role. From the reference count summary section, click the **Identities** to view a list of identities having access to the associated policy. Select the **View details** link to further browse each identity.
Identity Collections	Get count and access information on resources associated with this identity collection. From the reference count summary section, click the resource count to view resource details, such as its name, type and number of identities having access to this resource. You can browse further to view details of each resource.
Policies	Get count and access information on policies associated with this identity collection. From the reference count summary section, click the Policies count to view its details, such as its name, provider (OCI or Oracle Access Governance). owner, and so on. You can browse further to view association and reference details of each policy.
Resources	Get count of resources associated with this policy and resource details. From the reference count summary section, click the resource count to view resource details, such as its name, type and number of identities having access to this resource. You can browse further to view details of each resource.

7

Access Reviews

Access Reviews in Oracle Access Governance - Certify Access Privileges with Campaigns and Event-Driven Micro Certifications

Access Reviews, also known as *Access Certification* or *Access Attestation*, is the process to evaluate and certify the access privileges granted to identities within an enterprise. It checks and certifies if privileges granted are still required and align with the current job at work. Use the Oracle Access Governance **Access Reviews** feature for reviewing the access privileges. Make swift and accurate review decisions by examining insights and AI-powered recommendations based on prescriptive analytics.

Enterprises have large distributed landscape across on-premises and cloud systems. To avoid excessive accumulation of irrelevant permissions, or unauthorized access to critical information, *Access Reviews* are run regularly to monitor and certify the accesses. These are vital for secure access management and compliance processes.

Key Benefits of performing Access Reviews with Oracle Access Governance

Access Reviews help enterprises to govern frequent access changes, reduce cost, manage identity access lifecycle, maintain compliance, and strengthen the security posture. Enterprises handle voluminous access changes on daily basis. Regular reviews can proactively detect and remove excessive permissions or irrelevant privileges.

Oracle Access Governance **Access Reviews** feature offers various types to review access privileges. For example, use **Campaigns** to launch a set of ad hoc or periodic access reviews, or use micro-certifications, which are based on event change, timeline change, or detection of an orphan account.

Access Reviews in Oracle Access Governance helps to:

- **Reduce cost** by providing recommendations to remove non-essential and unwanted license or resources. For example, revoking application access for employees who no longer need it.
- **Strengthen security posture** of an enterprise by regularly reviewing accesses, ensuring that right resources have been granted just enough accesses for their role. For example, Oracle Access Governance performs automatic micro-certifications to detect event changes (location change, department change), timeline changes, or detect orphaned accounts that pose a security threat.
- **Meet governance compliance and requirements** by maintaining periodic access review audit reports. For example, ensuring compliance with industry regulations and compliance laws, such as GDPR, HIPAA, SOX, through regular access review campaigns.
- **Simplify decision-making** by recommending AI/ML-driven access review insights, such as peer group analysis, outlier detection, or recommendation. For example, Oracle Access

Governance gives recommendation to revoke a high-risk privilege for an identity based on prescriptive analytics.

Types of Access Reviews Offered by Oracle Access Governance

With Oracle Access Governance, you can run ad hoc or periodic Access reviews **Campaigns** over a set of identities, groups, accounts, roles, policies, and permissions. You can even run ownership reviews to verify resource ownership. Use near-real-time micro-certifications to run automatic reviews on specific components based on occurrence of an event change, timeline change, or unmatched accounts.

Access Review Campaigns: Ad hoc or Periodic Access Reviews

Use **Campaigns** to initiate periodic or ad hoc access review process. These are **snapshot-based reviews**, capturing all the relevant access information at a given point of time, and then assessing and generating access reviews tasks. Any change made to the data after the campaigns are setup won't be reflected in these reviews.

The access review types and selection criteria depend on Orchestrated System.

- Identity Access Reviews
- Policy Reviews
- Identity Collection Reviews
- Resource Ownership Reviews

Identity Access Reviews

Identity access reviews refer to assessing access privileges for identities in your enterprise, where access to a specific resource is verified or validated. You can certify access for *Workforce* and *Consumers* identities by running the identity access reviews. Reviewers, who are active *Workforce* users, can accept or revoke the assigned privileges.

Define selection criteria based on:

- **Users (who has access?)**: Select a set of core and custom identity attributes.
- **Applications (what are they accessing?)**: Select services, applications, or cloud accounts.
- **Permissions (which permissions?)**: Select permissions that are assigned directly in your Managed System, or permissions provisioned within Oracle Access Governance. You can use this criteria to quickly certify privileges for all Orchestrated Systems based on the permissions ingested directly from the Managed System. These are also called "reconciled permissions." For more information on running reviews based on permissions, refer to Identity Access Reviews based on Permissions Assigned Directly in Managed Systems.

> **Note:**
>
> – For the Oracle Access Governance system, you can run reviews based on permissions assigned directly (`DIRECT`) or Access Bundles granted through request from the **Which Permissions?** tile. Permissions or accounts provisioned through policy are not eligible in this review.
>
> – You can provision OCI IAM groups and application roles by creating an access bundle. For Oracle Cloud Infrastructure (OCI) system you can review OCI access bundles from the **Which Permissions?** tile.
>
> For the Oracle Access Governance system, you can run reviews based on permissions assigned directly (`DIRECT`) or Access Bundles granted through request from the **Which Permissions?** tile. Permissions or accounts provisioned through policy are not eligible in this review.

- **Roles (which roles?)**: Select roles provisioned to identities.

You can also define the approval workflow to select the number of review levels, review duration, and reviewer details. For more information on creating identity access reviews, refer to Create Identity Access Review Campaigns.

Example: You may run campaigns to evaluate and certify if junior associates or contractors in your organization have access to critical permissions or restricted information.

Identity Access Reviews based on Permissions Assigned Directly in Managed Systems

You can certify identity accesses by running identity access reviews on the permissions ingested directly from the Managed System. These are also called "reconciled permissions." Reconciled permissions refer to inherent permissions that are provisioned directly in the Managed systems without provisioning these from Oracle Access Governance. Run these reviews by selecting the Oracle Access Governance system from the **Campaigns** page.

Using insights and recommendations, reviewers can take action to accept or revoke these permissions. However, to manage your accesses at a granular level, use Access Bundles to provision the permissions.

When you select a set of permissions from the **Which Permissions?** tile, the system generates review tasks for all the eligible identities having access to these permissions either directly or through a request (Access Bundles).

Here's what's included in the review:

- Identities with permissions granted directly with grant type *DIRECT*.
- Identities with permissions granted as part of Access Bundles with grant type *Request*.
- Identities with permissions granted through Roles with grant type *Request*.
- Accounts of the identities that are provisioned directly or requested.

> **Note:**
>
> Permissions or accounts provisioned through policy, or Oracle Identity Governance (OIG) and Oracle Cloud Infrastructure (OCI) identity accounts are not covered in this review.

The components eligible for review vary based on the selected criteria. For more details, refer to the table in Eligible Orchestrated System Types to Launch Access Review Campaigns.

Things to Remember

- Permissions assigned directly or Access Bundles granted through request are available for review in the **Which Permissions?** tile. The grant type must be *Request* or *DIRECT* for the permissions to be included in the reviews.
- If directly assigned permissions are associated with an Access Bundle, which is then provisioned to identities through *Request*, then you will see only the Access Bundle and not individual permissions. For roles provisioned to identities through *Request*, it shows up as a role under the **Which roles?** tile.
  Example: If *Read* and *Alter* permissions are included in an Access Bundle and provisioned to identities via *Request*, you won't be able to view and review these particular permissions. You may choose to review the Access Bundle.
- If an account contains permissions granted through policy, then no review task will be generated for that account.
  Example: If an account contains four permissions, two of which are granted via policy, the review of permissions won't generate the account review task.

Scenario

You want to run access reviews for the identities based on *Read* and *Update* database permissions. Let's consider the following scenario:

- *Alice* has access to *Read* and *Update* permissions assigned directly.
- *Jane* has access to these permissions as part of Access Bundle, with additional *Write* permission.
- *Betty* being the database administrator has access to these permissions.

Here's what's included:

- For *Alice*, selected permissions will be reviewed.
- For *Jane*, Access Bundle will be reviewed with *Read* and *Update* and *Write* permissions only if Access Bundle is granted through **Request**.
- For *Betty*, role will be reviewed only if the role is granted through **Request**.
- *Alice*, *Jane*, and *Betty* identity accounts will be considered for review if the account or permissions associated with the account have not been provisioned using policy.

Reviewers can validate these access reviews from the **My Access Reviews → Identity** page following the process defined in Perform Access Reviews.

Remediation Actions

As part of the closed-loop access remediation process,

- If a reviewer revokes the permission, it will be revoked from the Managed system.
- If a permission is part of an Access Bundle granted to the identity, you won't be able to revoke that single permission unless you revoke an entire Access Bundle.

- Accounts associated with permissions are not revoked if accounts contain permissions granted through policy.
- Identity accounts are revoked only when all the permissions associated with accounts are revoked.

Policy Reviews

Review of Oracle Access Governance policies and OCI Identity and Access Management (IAM) policies to evaluate its effectiveness and compliance.

In Oracle Access Governance, you can create on-demand policy reviews, where you define the selection criteria to review policies. You can also define the approval workflow to select the number of review levels, review duration, and reviewer details. For more information, refer to Create Policy Review Campaigns and Types of Certification Tasks in Oracle Access Governance.

Example: You may run quarterly reviews on the defined network and storage policy of your tenancy to assess if these meet the principle of least privilege and applicable regulatory requirements.

Identity Collection Reviews

Membership review of a group to verify if only eligible set of members are assigned to a group. This is commonly known as "Group membership reviews."

You can create identity collection reviews for:

- Identity collections created in Oracle Access Governance
- OCI groups derived from Oracle Cloud Infrastructure (OCI)

*ExampleDatabase AdministratorSales Analyst*Create Identity Collection Review CampaignsTypes of Certification Tasks in Oracle Access Governance

Resource Ownership Reviews

Review ownership of resources that created within Oracle Access Governance, either periodically or on an ad hoc basis. By performing this review, you can ensure accountability of resources lies only with the designated owners.

Currently, you can run ownership reviews for the following Oracle Access Governance resources:

- Access Bundle
- Approval Workflows
- Identity Collections
- Orchestrated systems
- Policies
- Roles

Based on the approval workflow selected, the primary owner of a resource or any active workforce Oracle Access Governance identity will be considered for review. Reviewers can certify or change ownership of resources while performing reviews. For more information, refer to Create Ownership Reviews, and Resource Ownership Review Task.

Event-Based Micro-Certifications

Use **Event-Based Setup** to configure automated micro-certifications, triggered only when there are changes in the system of record, occurrence of an important date or time milestone, or detection of an orphan account. These are near **real-time reviews** and Oracle Access Governance continuously monitors profile changes to launch access reviews.

You must configure the attributes for which you want to enable event-based reviews. For more details, see Manage Identity Attributes.

As an active workforce user, you can review the event review tasks from the **My Access Reviews → Identity** page. If there's a change in event but no reviewable access items for that identity, then no review tasks will be generated for an identity.

You can configure event-based reviews for:

- **Change event**: Triggered by changes made in the identity profile, whenever an identity attribute is updated in the record system. These can be Core or Custom attributes.
- **Timeline event**: Triggered on the occurrence of a particular date, such as the work anniversary date of an employee to perform access review.
- **Unmatched event**: Triggered when an onboarded account doesn't match any identity within Oracle Access Governance.

Eligible Orchestrated System Types to Launch Access Review Campaigns

The type of access reviews and what you can review in Oracle Access Governance depends on the system type chosen while running reviews. You can review access to systems managed by Oracle Access Governance including ownership reviews, review access for Oracle Cloud Infrastructure (OCI), and review access to systems managed by Oracle Identity Governance (OIG).

Review Access to Systems Managed by Oracle Access Governance

Select this system to **run access reviews** or run **resource ownership reviews**. In the Oracle Access Governance system, you can run identity access reviews for all the orchestrated systems managed by Oracle Access Governance, such as Oracle Database, Flat File, Microsoft Active Directory, and so on.

We can broadly divide review requests into

- **Review Access**: To run identity access reviews, policy reviews, and identity collection reviews.
- **Review Ownership**: You can verify if only authorized owners are managing resources by running ownership reviews

Identity Access Reviews

Run identity access reviews for all the orchestrated systems using the Oracle Access Governance system. You can run these reviews based on core or custom identity attributes, applications they have access to, permissions granted from the Managed System, permissions provisioned within Oracle Access Governance as part of the Access Bundle, or roles granted to identities. For more information on running identity access reviews, refer to Identity Access Reviews.

Selection Criteria	Eligible Components in Review
Who has access? to select identity attributes	• Permissions • Access Bundles • Roles • Accounts
What are they accessing? to access applications	• Permissions • Access Bundles • Roles • Accounts
Which permissions? to select Access Bundles `(REQUEST)`	• Access Bundle • Roles • Accounts
Which permissions? to select Permissions `(DIRECT)`	• Permissions • Access Bundles • Roles • Accounts
	Note: If the account associated with the permissions is also included to give access through an Oracle Access Governance policy, then the account review tasks will not be generated.
Which roles? to select Roles `(REQUEST)`	Roles

Policy Reviews

Run Oracle Access Governance policy reviews to evaluate policy effectiveness. Policies created within Oracle Access Governance are reviewed. Oracle Access Governance policies contains details of resources and permissions attached to a group of identities by the means of roles and/or access bundles.

Identity Collection Reviews

Run membership review of a group to verify if only authorized set of members are assigned to a group. This is commonly known as "Group membership reviews." You can run reviews for identity collections created in Oracle Access Governance.

Review Ownership

You can verify if only authorized owners are managing resources by running ownership reviews. For example, you may run ownership reviews to verify if only designated and authorized owners are managing the Approval workflows.

Review Accesses for Cloud Services Managed by Oracle Cloud Infrastructure (OCI)

Select this system to certify identity access reviews, policy reviews, and OCI IAM identity collection membership reviews. Further, you can review assignment of OCI IAM groups and application roles managed by Oracle Access Governance.

The following review types are available:

- **Identity Reviews (Direct)**: Identity access rights by reviewing their application access, granted application roles.
- **Identity Access Reviews for Accesses Managed by Oracle Access Governance**: Identity access rights of OCI IAM groups and cloud services application roles assigned within Oracle Access Governance through access request.

> **Note:**
>
> If you request and assign Oracle Access Governance roles which includes the associated access bundles containing OCI IAM Groups or cloud services application roles, then choose to review Oracle Access Governance roles in the Oracle Access Governance system.

- **Policy Reviews**: OCI IAM policies that evaluates construct and functioning of a policy.
- **Identity Collection Reviews**: Group Memberships which evaluates that only eligible members have access to the group. If you choose to review OCI IAM groups and it contains a few members assigned from Oracle Access Governance, then with this review, you can only accept or revoke directly assigned members. For members assigned from Oracle Access Governance, choose to review the OCI Access Bundles using the **Which permissions?** tile.

Eligible Selection Criteria for Review in the Oracle Cloud Infrastructure (OCI) System

The components eligible for review vary based on the selected criteria, as follows:

Table 7-1 Access Review Selection Criteria in the OCI System

Selection Criteria	Eligible Components in Review
Which tenancies to select tenancies, cloud compartment, or domain	• OCI Accesses Managed by Oracle Access Governance • Roles • Identity Collections • Policies **Note:** If you opt to refine further to select a specific domain, the service won't generate policy reviews.
Who has access? to select identity attributes	• OCI Accesses Managed by Oracle Access Governance • Roles

Table 7-1 (Cont.) Access Review Selection Criteria in the OCI System

Selection Criteria	Eligible Components in Review
What are they accessing? to access cloud services	• OCI Accesses Managed by Oracle Access Governance • Roles
Which permissions? to select Access Bundles `(REQUEST)`	• Accesses Managed by Oracle Access Governance. It contains OCI Access Bundles
Which roles? to select application roles in OCI `(DIRECT)`	Roles assigned Directly in OCI.
Which policies? to select OCI Policies	Policies
Which identity collections? to select OCI IAM groups `(DIRECT)`	Identity Collections

Note:

If you choose to review OCI IAM groups and it contains a few members assigned within Oracle Access Governance, then with this review, you can only accept or revoke directly assigned members. For assignments managed by Oracle Access Governance, choose to review the OCI Access Bundles using the **Which permissions?** tile.

Review Access to Systems Managed by Oracle Identity Governance (OIG)

Select this system to certify access rights of an OIG identity by reviewing their application access, granted roles or permissions (entitlements). You cannot combine review for a specific permission (entitlement) and role in a single campaign.

Usage Examples: Certifying Access Privileges with Access Review Campaigns and Event-based Reviews

Let's see some of the scenarios where campaigns and automated access reviews are useful.

Example 1: Review Access Permissions for High-Profile Applications with Critical Functions

Scenario: To help your enterprise deter any harm against misuse of access rights for data sensitive applications, you need to schedule quarterly campaigns to certify access to critical functions, such as *update* and *terminate* permissions.

To do so, first select the system, then apply filters to select data sensitive applications using the **What are they accessing?** tile. Select appropriate permissions using **Which permissions?**. Complete the campaign steps to assign appropriate workflow and campaign details. Post this, the review tasks will be generated that the reviewer can review on the **My Access Review** page.

> **Note:**
>
> You can either create a campaign to review permissions or review roles but both cannot be selected in a single campaign. In this example, **Which roles?** will be disabled along with **Which policies?** As we selected to review identity access, policy and identity collection review selection parameters will also be disabled.

Example 2: Review Policies for all Cloud Resources

Scenario: Your company updated their security protocols for data storage. As a cloud security administrator, you need to carry out on-demand access reviews of all the IAM policies available in your cloud account to ensure that it meets the latest security standards and regulations.

To do so, first select the system, add selection criteria to select cloud provider, cloud account, domain, or compartment. Complete the campaign steps to assign appropriate workflow and campaign details. In this example, the campaign reviewer can review all the applicable review tasks on the **My Access Reviews → Access Control** tab, with type **Policy**.

Example 3: Group Membership Reviews for Project Groups

Scenario: As a project manager, carry out quarterly group membership review for your team to ensure only current team members have access to code repositories and access to required third-party applications. This process will help you to remove any unauthorized access as well as monitor and control project cost.

To do so, first select the OCI system, add selection criteria to select OCI groups, add an approval workflow along with campaign details. In this example, the campaign reviewer can review all the applicable review tasks on the **My Access Reviews → Access Control** tab, with type **Identity Collection**.

Example 4: Enabling Automated Access Reviews for Employees Triggered by a Change Event

Scenario: As a business owner, you need to set up automated access reviews to perform micro-certifications whenever manager, job code, or location changes for an employee.

For this, enable event-based access reviews for job-code, manager, and location. Whenever the latest data synchronization happens from the orchestrated system with these updates, Oracle Access Governance will automatically raise multiple event-based access reviews associated with this single identity.

Access Review Campaigns

Working with Access Review Campaigns

Use *Campaigns* to initiate an access review process. To use Access Reviews effectively, understand the campaign lifecycle, along with crucial concepts, such as self-certification of accesses and fallback mechanism when an invalid reviewer or owner is detected. Use

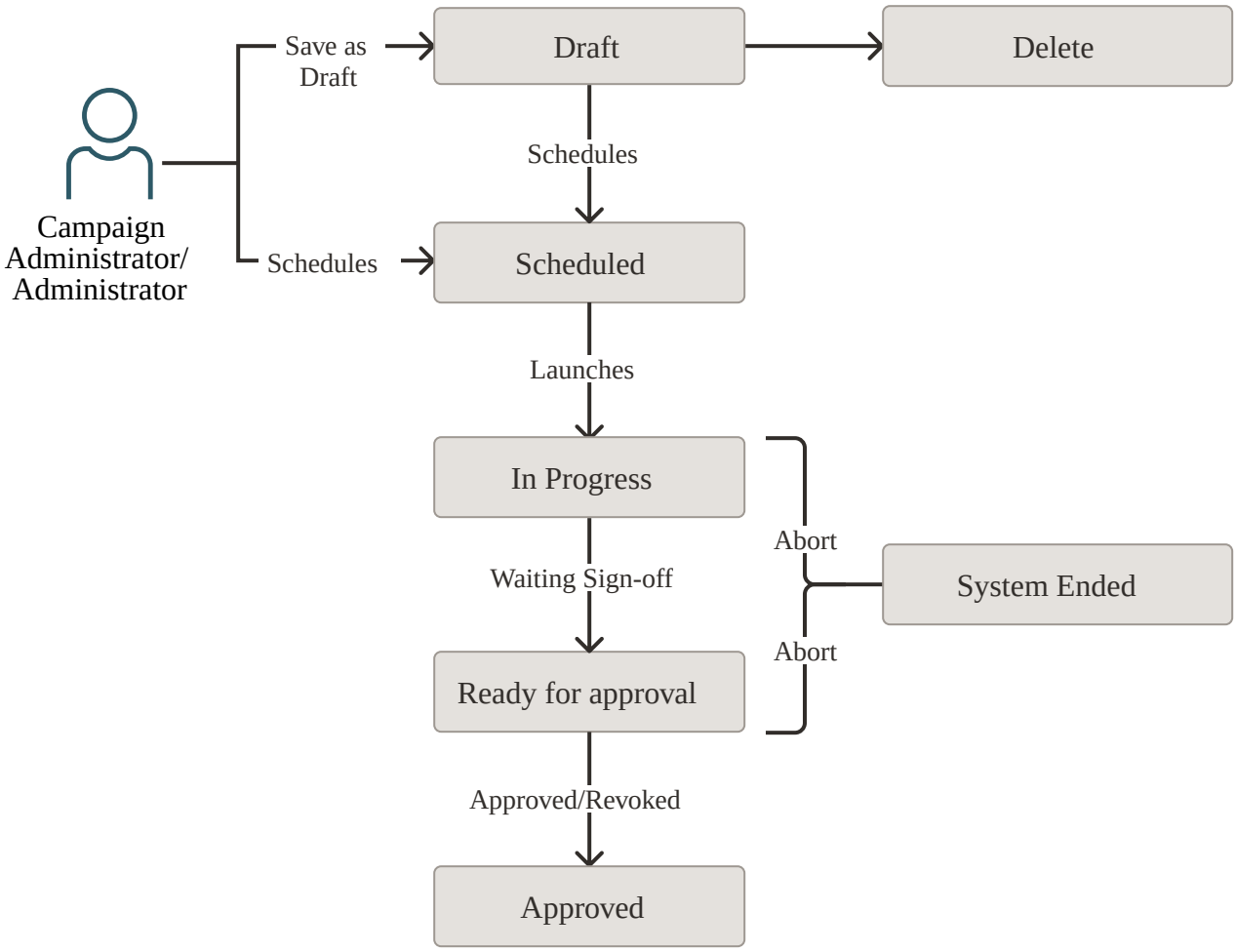
guidelines or best practices while working with campaigns to ensure effective review process is conducted.

Access Review Campaign Stages

As an *Administrator* or *Campaign Administrator*, to certify access privileges, first set up and schedule Access Review Campaigns. During its lifecycle, a campaign courses through various access review states. The tasks that you can perform depend on the state or the status of a campaign.

As an *Administrator* or *Campaign Administrator*, initiate the access review process by creating a **Campaign** from the **Access Reviews** section. You can set up either an ad-hoc campaign or schedule a periodic campaign, forming a campaign series. A campaign proceeds through various stages or states in its lifecycle. This involves defining the scope, setting approval workflows, selecting campaign owners, and scheduling campaigns. Once launched, reviewers can accept or revoke access privileges. The decisions taken are fulfilled as part of the closed-loop remediation process.

Here are the certification states for Access Review Campaigns:



- **Draft**: When a new Access Review Campaign is created or added but not yet launched. In the *Draft* state, you can:
  - View campaign details

  - Edit a campaign
  - Delete a campaign
- **Scheduled**: When an access review campaign is created to be launched at a specific time in future. In the *Scheduled* state, you can:
  - View campaign details
  - Edit a campaign
  - Clone a campaign
  - Terminate a campaign
  - Terminate Campaign Series
- **In Progress**: When an access review campaign is launched. Campaign reviewers are notified about the campaign over email. Reviewers can make decisions on the assigned review tasks by accepting or revoking access privileges to finally fulfilling the decision as part of the closed-loop remediation process. In an *In Progress* state, you can:
  - View campaign details
  - Clone a campaign
  - Terminate a campaign
  - Terminate Campaign Series
  - View report
  - Download CSV data
- **Ready for Approval**: When the review tasks have been completed or the campaign due date has elapsed, the campaign moves to the **Ready for Approval** state. In case, there are pending review items, the suggested actions given in the approval workflow are automatically considered. For example, approve all unreviewed access review tasks. In the **Ready for Approval** state, you can:
  - View campaign details
  - Clone a campaign
  - Terminate a campaign
  - Terminate Campaign Series
  - View report
  - Download CSV data
- **Approved**: When a campaign owner approves and sign-off a campaign from the *Actions* option, it is marked as **Approved**. The campaign moves from the **my ongoing campaigns** queue to the **my previous campaigns** queue. In the **Approved** state, you can:
  - View campaign details
  - Clone a campaign
  - View report
  - Download CSV data
- **System Ended**: When an unexpected error occurs, the campaign may be aborted leading to the **System Ended** status. In the *System Ended* status, you can view campaign details, clone a campaign, view report, or download the CSV report. A few possible causes are:
  - When an internal system error occurs, such as failure in generating Insights or failure in validating campaign criteria.

  - All *Draft* and *Scheduled* campaigns created before June 2023 release are automatically aborted and marked as *System ended*.
  - When an Oracle Access Governance service instance is deleted, all the campaigns in that service instance are aborted and marked as *System Ended*.
  - When a system failure occurs during termination of a campaign, the campaign is aborted and results in the *System Ended* state.

- **Terminated**: When a campaign is terminated by a *Campaign Administrator* or a Campaign owner. You can terminate a campaign when it is in the *Scheduled*, *In Progress*, or *Ready for Approval* state. A campaign is also terminated when the:
  - Reviewer is inactive and managerial hierarchy does not have an active user, or the campaign owner is inactive.
  - Fallback process fails to assign an appropriate campaign owner or reviewer, the campaign is *Terminated* by the system.
  - Number of members in the Identity Collection is fewer than the defined reviewers for the Identity Collection approval worlflow.

  In the *Terminated* state, you can:
  - View campaign details
  - Clone
  - View report
  - Download CSV data

Understanding Self-Certification Guardrails

Self-certification is a process of approving or certifying your own access rights without the intervention of an external reviewer. It is a valid business process established to reduce the administrative burden or for other appropriate business justifications. However, self-certification is usually not recommended for high-risk accesses involving critical data, or where a potential personal benefit is involved. Oracle Access Governance gives you the option to either enable or disable the self-approval process.

Based on the approval workflow type, Oracle Access Governance enables or disables the self-certification guardrails for a Campaign.

- If you select **Custom User**, **Identity Collection**, or **Owner** workflow, then you can choose to enable or disable the self-certification process. If you choose the **Beneficiary** workflow, then also, you can self-approve your accesses.
- If you select any other workflow or choose to disable the self-approval process, then system initiates an appropriate fallback mechanism to auto-assign the review task to the next available valid reviewer.

Understanding Fallback Mechanism: Methods to Prevent Campaign Termination

While working with Campaigns, you choose your intended reviewer by selecting one of the approval templates defined in the Oracle Access Governance **Approval Workflows** feature. The **Campaigns** service will initiate a fallback mechanism in case an invalid reviewer or an invalid campaign owner is detected to prevent termination of a campaign.

Here's when Oracle Access Governance tags a reviewer as invalid:

- When an **Inactive** Oracle Access Governance identity is selected as a reviewer.
- When an active identity with the **Consumer** user type is selected as a reviewer.

- When self-approval is disabled in the selected approval template, and the reviewer is same as the beneficiary whose accesses are being reviewed or certified.

Fallback Mechanism for an Invalid Reviewer

If the intended reviewer is invalid, then Oracle Access Governance initiates the following fallback mechanism, in the order listed, to assign a valid reviewer:

Intended Reviewer → Management Chain of the intended reviewer → Campaign owner → Any user, randomly selected having the Access Governance Administrator role.

- Intended Reviewer
- Immediate manager of the reviewer, up to the defined management chain until a valid reviewer is found.
- If no active managers are found, then the reviewer is set as Campaign owner.
- If self-approval is not allowed, no active managers are found, campaign owner is the beneficiary, then any one user, chosen randomly, with the *Administrator* roles is automatically assigned as an access review reviewer.

Fallback Mechanism for an Invalid Campaign Owner

Invalid campaign owners can be inactive users, consumer users, or users not part of the approval workflow.

If the intended campaign owner is invalid, then Oracle Access Governance initiates the following fallback mechanism to assign a valid campaign owner:

Intended Campaign Owner → Management Chain of the campaign owner → Any user, randomly selected, having the Access Governance Administrator role.

- Management chain of the campaign owner.
- If no valid managers are found, then any user, chosen randomly, with the *Administrator* role is automatically assigned as the campaign owner.

Example 1 - Understanding Fallback Mechanism when Self-Certification is Allowed

Scenario: As a *Campaign Administrator* and *Campaign Owner*, Sarah launches the periodic identity access reviews for her own department. She selects the *Owner* approval workflow template and allows self-approval of access reviews. It generates two access reviews, with the Assignment type **Account** as follows:

- *Beneficiary*: John Doe and *Account owner* as Sarah
- *Beneficiary*: Sarah and *Account owner* as Sarah

Using the *Owner* template with self-certification enabled, the intended reviewer for this campaign will be the account owner, as follows:

- *Beneficiary* as John Doe and *Reviewer* as Sarah
- *Beneficiary* as Sarah and *Reviewer* as Sarah

Example 2 - Understanding Fallback Mechanism when Self-Certification is Not Allowed

Scenario: As a *Campaign Administrator* and *Campaign Owner*, Sarah launches the ad-hoc identity access reviews for critical functions in high-risk applications for her department. She selects the *Owner* approval workflow template and does not allow self-approval of access reviews. It generates two access reviews, with the Assignment type **Permission** as follows:

- *Beneficiary*: John Doe and *Permission owner* as Sarah

- *Beneficiary*: Sarah and *Permission owner* as Sarah

As the self-certification is disabled, the intended reviewer for this campaign cannot be same as the beneficiary. So, fallback mechanism will be initiated as follows:

Intended Reviewer → Management Chain of the intended reviewer → Campaign owner → Any user, randomly selected having the Access Governance Administrator role.

Assume that no valid manager is found in the management chain, then next campaign owner should be assigned as a reviewer. In this example, campaign owner is same as the beneficiary with self-certification disabled, then the access reviewer is chosen randomly, having the *Administrator* role, which in this example is *Carol Beck*. So access reviewers will be as follows:

- *Beneficiary* as John Doe and *Reviewer* as Sarah
- *Beneficiary* as Sarah and *Reviewer* as Carol Beck

Best Practices: Guidelines to Consider While Working With Campaigns

While running campaigns, you must adhere to a few best practices and guidelines to ensure effective access review process.

Here are a few guidelines you must adhere to while running campaigns:

- Campaigns can only be created by Oracle Access Governance *Administrator* or *Campaign Administrator* .
- All campaigns can only be managed by Oracle Access Governance *Administrator*. *Campaign Administrator* can manage the campaigns that they created. Campaign owners can manage the campaign they own.
- You can run identity reviews based on permissions granted directly in your Managed Systems (also known as reconciled permissions) without the need to provision it from Oracle Access Governance. However, to manage your accesses at a granular level, use Access Bundles and provision the permissions from Oracle Access Governance.
- You can quickly certify privileges by running identity access reviews from the Oracle Access Governance system based on the permissions assigned directly. Permissions or accounts provisioned through policy, or Oracle Identity Governance (OIG) and Oracle Cloud Infrastructure (OCI) identity accounts are not covered in this review. For more information on running reviews based on reconciled permissions, refer to Identity Access Reviews for Permissions Assigned Directly in Managed Systems.
- In a single campaign, you cannot combine two different types of access reviews. For example, if you create a campaign to review policies, criteria for identity access reviews or identity collection reviews are no longer applicable and are disabled.
- Campaigns can be certified by any active user associated with a specific approval workflow. Reviewers can view only their associated review tasks. A reviewer who is not associated with any approval workflow cannot perform tasks against any reviews.
- If no reviews are generated, it will automatically proceed to **Ready for approval** state.
- Campaign Owner:
  - must be an Oracle Access Governance active user.
  - can receive email notifications whenever a campaign progresses through various campaign states.
  - can be an access review reviewer based on the fallback mechanism if the original intended reviewer is invalid.
  - Can manage a campaign that they own.

- You can self-approve or self-certify your accesses using the **Custom user**, **Identity Collection**, or **Owner** template. You can also self- approve your accesses when using the **Beneficiary** approval template.
- You can certify access privileges for consumer users, but a consumer user cannot be an access reviewer.
- For policy reviews, you must not modify the policy after the campaigns have been scheduled. It will result in failure of completing the remediation request. The policy statements should be consistent throughout the campaign process.
- For identity collection reviews, you must not modify members after the campaigns have been scheduled. It will result in failure of completing the remediation request. The list of members should be consistent throughout the campaign process.

Create Identity Access Review Campaigns

As an *Administrator* or *Campaign Administrator*, certify identity accesses by creating on-demand Identity Access Review campaigns from the Oracle Access Governance Console. These can be one-time or periodic access review campaigns.

Prerequisites

Before you create identity access review campaigns, consider the following:

- To create campaigns for access reviews, you must have Oracle Access Governance *Administrator* or *Campaign Administrator* role assigned to you.
- Enable the identity attributes, both core and custom, from the **Identity Attributes** page. For example, you may need to define your campaigns based on *Project Code* or *Cost Center*. See View and Configure Custom Identity Attributes.
- You must select at least one selection criteria to run Campaigns.
- Choose the Oracle Access Governance system to run identity access reviews based on the permissions ingested directly from the Orchestrated systems.
- For the Oracle Access Governance system, you can choose permissions assigned directly (`DIRECT`) or Access Bundles granted through request from the **Which Permissions?** tile. Permissions or accounts provisioned through policy are not eligible in this review.
- You cannot review specific permissions and roles in a same campaign as **Which permissions?** and **Which roles?** are mutually exclusive. This means that you can select either of the two while creating a campaign. However, you can review all the available permissions and roles when you select **Who has access?** and **What are they accessing?**.

For more information on Campaigns, see Best Practices to work with Campaigns.

Navigate to Campaigns

Campaigns are created from the Oracle Access Governance Console. Go to **Campaigns** page to launch the on-demand access review process.

1. Log in to the Oracle Access Governance Console.
2. Select one of the following options to navigate to the screen:
   - On the Oracle Access Governance Console home page, select the **Access Reviews** tab. Select the **Define a new campaign** tile.

- From the **Navigation Menu**, select **Access Reviews**, and then **Campaigns**. From the **Campaigns** page, select **Create a campaign**.

3. Select one of the system types for which you want to run Campaigns. For more information, see Eligible System Types to Launch Access Review Campaigns.

   On the **Create a new access review campaign** workflow page, define the selection criteria for your campaign.

Select Criteria for your Access Reviews

In the **Selection criteria** dimension, you select appropriate criteria for your *Identity Access Review* Campaigns. The attributes configured in the **Identity Attributes** page are available as the selection criteria. All criteria can be searched by *name*.

The selection tiles are based on the system selected in the previous step. For example, for Oracle Cloud Infrastructure, you may see additional tiles, like **Which tenancies?** so that you can select your cloud account for which you want to run review.

1. Select one or more criteria tiles that you wish to include in any order. You don't need to update each criteria. The selection values are derived from the integrated orchestrated system. Available tiles are:

Option	Description
Who has access?	To filter identities based on core or custom identity attributes. a. Select up to five attributes in the **Which attributes do you want to add for selection?** field. b. From each tab, select one or more available selection values.
What are they accessing?	To filter identities based on their access to applications or resources.
Which permissions?	To filter identities based on their access to permissions. • For Oracle Identity Governance (OIG), you can select entitlements. • For Oracle Access Governance, you can select permissions assigned directly in the Managed System or permissions provisioned through Access Bundle via *Request* within Oracle Access Governance . The permissions vary based on the orchestrated system. • For Oracle Cloud Infrastructure (OCI), you can review OCI IAM Groups and Application Roles assigned through Access Bundle via *Request* within Oracle Access Governance.

Option	Description
Which roles?	To filter identities based on their roles. • For Oracle Identity Governance (OIG), you can select directly assigned roles. • For Oracle Access Governance, you can select roles assigned directly in the Managed System or created within Oracle Access Governance. • For Oracle Cloud Infrastructure (OCI), you can review OCI Cloud services application roles assigned directly in OCI.
Which tenancies?	To filter cloud account. Select the **Refine further** link to select compartment and domain for your cloud account. Available only for Oracle Cloud Infrastructure review system.

2. After selection, select **Apply my selections**.
3. To update your selection criteria, select the **Modify** button on the relevant tile.

   The panel on the right-side of the page shows you the effect of your selection and provides you with an estimate of included identities considered for review.
4. Once you've made your selection, select **I'm good, go to workflows** button to proceed to the *Assign workflow* dimension.

   At any point of time, select **Save draft** to save your campaign and pick up later to work on the details.

Add Access Reviewers by Selecting Approval Workflow

In the *Assign Workflow* dimension, you select the approval workflow for your access review.

1. Select which approval workflow you want to assign to this access review campaign.

   A list of the available workflows shows all approval workflows defined in your system. Consider Self Certification Guardrails and Fallback Mechanism in Access Review before selecting the workflow. For details on how to create and manage approval workflows see Create Approval Workflow and Manage Approval Workflow.
2. After you have selected your workflow, click the **View approval workflow** link to see a graphical representation of the selected workflow.
3. Select the scope of justification required for review decisions. You can select for reviewers to add comments for all the review decisions, for revoke decisions only, or keep the justification field as optional.
4. Select **Next** to proceed to the *Add details* dimension.

   At any point of time, select **Save draft** to save your campaign and pick up later to work on the details.

Add Campaign Details

In the *Add Details* dimension, select campaign schedule cycle, give a meaningful name to your campaign, add a supporting description, and assign values to additional attributes, such as campaign owner, and when the campaign should start or end.

To add details :

1. Select an appropriate schedule cycle in the **How often do you want this to run?** field.
2. In **What do you want to call this campaign?**, enter a unique campaign name.
3. In **How do you want to describe this campaign**, enter campaign description.
4. In the **Who owns this campaign?** field, select campaign owner.

   Consider Self Certification Guardrails and Fallback Mechanism before assigning the campaign owner.
5. Based on the schedule cycle selected in Step 1, select the time at which you want to launch the campaign.
   - For One-Time, select either **Run now** or **Schedule Later**. By default, the campaign is set to begin at the top of the next hour, the following day of campaign creation.
   - For campaign series, select the calendar icon and select the start and end date and time for the campaign.
6. Once you have set your preferences, select **Next** to go to the *Review and submit* dimension.
7. Optional: You may select one of the additional actions:
   - **Save Draft**: To save your changes and later come back and edit the workflow or details.
   - **Cancel**: To cancel the current process.
   - **Back**: To go back to the previous step.

Review and Submit the Campaign

In the *Review and submit* dimension, review the campaign details and create the campaign.

To review and submit your campaign :

1. Review the campaign information. For any changes, select the **Back** button.
2. Select **Create**. The campaign is successfully scheduled.

Create Policy Review Campaigns

As an *Administrator* or *Campaign Administrator*, certify policies by creating on-demand Policy Review campaigns from the Oracle Access Governance Console. These can be one-time or periodic policy review campaigns.

Currently, you can certify policies for systems managed by Oracle Cloud Infrastructure (OCI) and Oracle Access Governance.

Navigate to Campaigns

Campaigns are created from the Oracle Access Governance Console. Go to **Campaigns** page to launch the on-demand access review process.

1. Log in to the Oracle Access Governance Console.
2. Select one of the following options to navigate to the screen:
   - On the Oracle Access Governance Console home page, select the **Access Reviews** tab. Select the **Define a new campaign** tile.
   - From the  **Navigation Menu**, select **Access Reviews**, and then **Campaigns**. From the **Campaigns** page, select **Create a campaign**.
3. Select one of the system types for which you want to run Campaigns. For more information, see Eligible System Types to Launch Access Review Campaigns.

   On the **Create a new access review campaign** workflow page, define the selection criteria for your campaign.

Select Criteria for your Access Reviews

In the **Selection criteria** dimension, you select appropriate criteria for your *Policy Review* Campaigns. All criteria can be searched by *name*.

Currently, you can certify policies for systems managed by Oracle Cloud Infrastructure (OCI) and Oracle Access Governance.

1. Select one or more criteria tiles that you wish to include in any order. You don't need to update each criteria. The selection values are derived from the integrated orchestrated system. Available tiles are:

Option	Description
Which tenancies?	To filter and select cloud account. Select the **Refine further** link to select compartment and domain for your cloud account. Available only for Oracle Cloud Infrastructure (OCI) system.
Which Policy?	To filter and select policies which you wish to review. You can search specific policy by its name or add filters on policy's creation date to limit the scope of your search results.

2. After selection, select **Apply my selections**.
3. To update your selection criteria, select the **Modify** button on the relevant tile.

   The panel on the right-side of the page shows you the effect of your selection and provides you with an estimate of included policies considered for review.
4. Once you've made your selection, select **I'm good, go to workflows** button to proceed to the *Assign workflow* dimension.

Add Access Reviewers by Selecting Approval Workflow

In the *Assign Workflow* dimension, you select the approval workflow for your access review.

1. Select which approval workflow you want to assign to this access review campaign.

   A list of the available workflows shows all approval workflows defined in your system. Consider Self Certification Guardrails and Fallback Mechanism in Access Review before selecting the workflow. For details on how to create and manage approval workflows see Create Approval Workflow and Manage Approval Workflow.
2. After you have selected your workflow, click the **View approval workflow** link to see a graphical representation of the selected workflow.
3. Select the scope of justification required for review decisions. You can select for reviewers to add comments for all the review decisions, for revoke decisions only, or keep the justification field as optional.
4. Select **Next** to proceed to the *Add details* dimension.

   At any point of time, select **Save draft** to save your campaign and pick up later to work on the details.

Add Campaign Details

In the *Add Details* dimension, select campaign schedule cycle, give a meaningful name to your campaign, add a supporting description, and assign values to additional attributes, such as campaign owner, and when the campaign should start or end.

To add details :

1. Select an appropriate schedule cycle in the **How often do you want this to run?** field.
2. In **What do you want to call this campaign?**, enter a unique campaign name.
3. In **How do you want to describe this campaign**, enter campaign description.
4. In the **Who owns this campaign?** field, select campaign owner.

   Consider Self Certification Guardrails and Fallback Mechanism before assigning the campaign owner.
5. Based on the schedule cycle selected in Step 1, select the time at which you want to launch the campaign.
   - For One-Time, select either **Run now** or **Schedule Later**. By default, the campaign is set to begin at the top of the next hour, the following day of campaign creation.
   - For campaign series, select the calendar icon and select the start and end date and time for the campaign.
6. Once you have set your preferences, select **Next** to go to the *Review and submit* dimension.
7. Optional: You may select one of the additional actions:
   - **Save Draft**: To save your changes and later come back and edit the workflow or details.
   - **Cancel**: To cancel the current process.
   - **Back**: To go back to the previous step.

Review and Submit the Campaign

In the *Review and submit* dimension, review the campaign details and create the campaign.

To review and submit your campaign :

1. Review the campaign information. For any changes, select the **Back** button.

2. Select **Create**. The campaign is successfully scheduled.

Certify Group Memberships with Identity Collections Review Campaigns

As an *Administrator* or *Campaign Administrator*, certify group memberships by creating on-demand Identity Collections Review campaigns from the Oracle Access Governance Console. These can be one-time or periodic policy review campaigns.

Currently, you can certify group membership for systems managed by Oracle Cloud Infrastructure (OCI) and Oracle Access Governance. If you choose to review OCI IAM groups and it contains a few members provisioned from Oracle Access Governance, then with this review, you can only accept or revoke directly assigned members. For members provisioned from Oracle Access Governance, choose to review the OCI Access Bundles using the **Which permissions?** tile.

Navigate to Campaigns

Campaigns are created from the Oracle Access Governance Console. Go to **Campaigns** page to launch the on-demand access review process.

1. Log in to the Oracle Access Governance Console.
2. Select one of the following options to navigate to the screen:
   - On the Oracle Access Governance Console home page, select the **Access Reviews** tab. Select the **Define a new campaign** tile.
   - From the  **Navigation Menu**, select **Access Reviews**, and then **Campaigns**. From the **Campaigns** page, select **Create a campaign**.
3. Select one of the system types for which you want to run Campaigns. For more information, see Eligible System Types to Launch Access Review Campaigns.

   On the **Create a new access review campaign** workflow page, define the selection criteria for your campaign.

Select Criteria for your Access Reviews

In the **Selection criteria** dimension, you select appropriate criteria for your *Policy Review* Campaigns. All criteria can be searched by *name*.

Currently, you can certify group membership for systems managed by Oracle Cloud Infrastructure (OCI) and Oracle Access Governance. In one campaign, you can certify membership either for OCI groups or for Identity collections created within Oracle Access Governance. You cannot combine two different types of groups in one campaign.

1. Select one or more criteria tiles that you wish to include in any order. You don't need to update each criteria. The selection values are derived from the integrated orchestrated system. Available tiles are:

Option	Description
Which tenancies?	To filter and select cloud account. Select the **Refine further** link to select compartment and domain for your cloud account. Available only for Oracle Cloud Infrastructure (OCI) system.
Which identity collections?	To filter and select identity collections for which you wish to review the group

Option	Description
	membership. You can search specific policy by its name or add filters on policy's creation date to limit the scope of your search results.

2. After selection, select **Apply my selections**.
3. To update your selection criteria, select the **Modify** button on the relevant tile.

   The panel on the right-side of the page shows you the effect of your selection and provides you with an estimate of included policies considered for review.
4. Once you've made your selection, select **I'm good, go to workflows** button to proceed to the *Assign workflow* dimension.

Add Access Reviewers by Selecting Approval Workflow

In the *Assign Workflow* dimension, you select the approval workflow for your access review.

1. Select which approval workflow you want to assign to this access review campaign.

   A list of the available workflows shows all approval workflows defined in your system. Consider Self Certification Guardrails and Fallback Mechanism in Access Review before selecting the workflow. For details on how to create and manage approval workflows see Create Approval Workflow and Manage Approval Workflow.
2. After you have selected your workflow, click the **View approval workflow** link to see a graphical representation of the selected workflow.
3. Select the scope of justification required for review decisions. You can select for reviewers to add comments for all the review decisions, for revoke decisions only, or keep the justification field as optional.
4. Select **Next** to proceed to the *Add details* dimension.

   At any point of time, select **Save draft** to save your campaign and pick up later to work on the details.

Add Campaign Details

In the *Add Details* dimension, select campaign schedule cycle, give a meaningful name to your campaign, add a supporting description, and assign values to additional attributes, such as campaign owner, and when the campaign should start or end.

To add details :

1. Select an appropriate schedule cycle in the **How often do you want this to run?** field.
2. In **What do you want to call this campaign?**, enter a unique campaign name.
3. In **How do you want to describe this campaign**, enter campaign description.
4. In the **Who owns this campaign?** field, select campaign owner.

   Consider Self Certification Guardrails and Fallback Mechanism before assigning the campaign owner.
5. Based on the schedule cycle selected in Step 1, select the time at which you want to launch the campaign.
   - For One-Time, select either **Run now** or **Schedule Later**. By default, the campaign is set to begin at the top of the next hour, the following day of campaign creation.

- For campaign series, select the calendar icon and select the start and end date and time for the campaign.

6. Once you have set your preferences, select **Next** to go to the *Review and submit* dimension.
7. Optional: You may select one of the additional actions:
   - **Save Draft**: To save your changes and later come back and edit the workflow or details.
   - **Cancel**: To cancel the current process.
   - **Back**: To go back to the previous step.

Review and Submit the Campaign

In the *Review and submit* dimension, review the campaign details and create the campaign.

To review and submit your campaign :

1. Review the campaign information. For any changes, select the **Back** button.
2. Select **Create**. The campaign is successfully scheduled.

Create Ownership Review Campaigns

You can review ownership of resources that are created within Oracle Access Governance by setting up on-demand Ownership Review Campaigns. These can be one-time or periodic review campaigns.

Navigate to Review Ownership Campaigns

Ownership Campaigns are created from the Oracle Access Governance Console. Choose the **Oracle Access Governance** system to launch the on-demand ownership review process.

1. Log in to Oracle Access Governance Console.
2. Select one of the following options to go to the **Campaigns** page:
   a. On the Oracle Access Governance Console home page, select the **Access Reviews** tab. Select the **Define a new campaign** tile.

   b. From the  Navigation Menu, select **Access Reviews**, and then **Campaigns.**. From the **Campaigns** page, select **Create a campaign**.
3. Select the Oracle Access Governance review system, and then select **Review ownership**.

On the **Create a new ownership review campaign** workflow page, choose the resources for which you want to run the ownership reviews.

Choose Resources for Ownership Reviews

Select the resources for which you want to review ownership. By default, Oracle Access Governance considers ownership review for all the resources.

1. Select the resource tile for which you want to review ownership. Clear the tile that you don't want to include in the review. You can choose from:
   - Access Bundles
   - Approval Workflows

- Identity Collections
- Orchestrated System
- Policies
- Roles

2. Click **Next** to refine your selection and apply filters in the **Add filters** step.

   At any point of time, select **Save draft** to save your campaign and pick up later to work on the details.

Apply Filters to Select Resources

Add filters to refine your selection. You can choose to review the ownership of resources that haven't been reviewed within the selected number of days. You can refine further to choose your specific resources by adding filters. By default, all the available resources are considered for ownership review.

1. Add Filter in the **Last Reviewed** step to select the number of days since the last ownership review.

   Example: Choose **Over 90 days** to include ownership review of resources that haven't been reviewed in the last three months or 90 days.

2. Refine your selection in the **Selection criteria** step.
   a. Select the **Add Filter** button.
   b. Select one of the resources in the **What do you want to refine?** list.
   c. Select the search criteria to find your resources. You can search a resource by resource name, primary owner name, last updated, created date, or created by options.
   d. Select the logical operator of your choice.
   e. Select the value and click **Apply**.
3. In the **Selection filter** section
   a. Select the  edit icon to modify the value listed in the applied filter.
   b. Select the  delete icon to remove the filter.

   The panel on the right-side of the page shows you the effect of your selection and provides you with an estimate of included resources considered for ownership review. You can also view current filters selected for the campaign.
4. Click **Next** to proceed to the **Assign workflow** step.

Add Access Reviewers by Selecting Approval Workflow

In the *Assign workflow* dimension, you select the approval workflow for your access review.

1. Select one of the following approval workflows to assign to the access review campaign.
   a. **Owner**, where primary owner of a resource will be assigned as the reviewer. Consider fallback process before configuring this workflow.
   b. **Custom User**, and then select any active workforce Oracle Access Governance identity as a reviewer.

2. After you have selected your workflow, click the **View approval workflow** link to see a graphical representation of the selected workflow.
3. Select the scope of justification required for review decisions. You can select for reviewers to add comments for all the review decisions or keep the justification field as optional.
4. Select **Next** to proceed to the *Add details* dimension.

   At any point of time, select **Save draft** to save your campaign and pick up later to work on the details.

Add Campaign Details

In the *Add Details* dimension, select campaign schedule cycle, give a meaningful name to your campaign, add a supporting description, and assign values to additional attributes, such as campaign owner, and when the campaign should start or end.

To add details :

1. Select an appropriate schedule cycle in the **How often do you want this to run?** field.
2. In **What do you want to call this campaign?**, enter a unique campaign name.
3. In **How do you want to describe this campaign**, enter campaign description.
4. In the **Who owns this campaign?** field, select campaign owner.

   Consider Self Certification Guardrails and Fallback Mechanism before assigning the campaign owner.
5. Based on the schedule cycle selected in Step 1, select the time at which you want to launch the campaign.
   - For One-Time, select either **Run now** or **Schedule Later**. By default, the campaign is set to begin at the top of the next hour, the following day of campaign creation.
   - For campaign series, select the calendar icon and select the start and end date and time for the campaign.
6. Once you have set your preferences, select **Next** to go to the *Review and submit* dimension.
7. Optional: You may select one of the additional actions:
   - **Save Draft**: To save your changes and later come back and edit the workflow or details.
   - **Cancel**: To cancel the current process.
   - **Back**: To go back to the previous step.

Review and Submit the Campaign

In the *Review and submit* dimension, review the campaign details and create the campaign.

To review and submit your campaign :

1. Review the campaign information. For any changes, select the **Back** button.
2. Select **Create**. The campaign is successfully scheduled.

Manage and Monitor Access Review Campaigns

Users with the *Administrator* application roles can manage and monitor all types of access review campaigns using the Oracle Access Governance Console.

Users with the following Oracle Access Governance roles can access these:

- *Administrator*: Can manage and monitor all campaigns in the Oracle Access Governance Console.
- *Campaign Administrator*: Can manage and monitor campaigns that they have created.
- *Auditor*: Can monitor all campaigns in Oracle Access Governance.
- *Campaign Owner (User)*: Can manage and monitor the campaigns that they own.

Search and Apply Filters to View Available Campaigns

You can retrieve campaign by performing keyword search or applying filters based on campaign status. You can apply suggested filters to view focused information.

Campaign Owners can only view the campaigns that they own, and not all the available campaigns.

1. Log on to the Oracle Access Governance Console with a user assigned either the *Administrator* or *Campaign Administrator* application role.
2. Select one of the following options to navigate to the **Campaigns** page:
   - On the console home page, select the **Access Reviews** tab and then select one of the following options.
     - To view in *In progress* or *Ready for Approval* campaigns, select the **Show me my <number> ongoing campaigns** tile.
     - To view *Ready for Approval* campaigns, select the **<number> campaigns are ready for approval** tile.
     - To view all campaigns, select the **Show me all campaigns** tile.
   - Click the  **icon**, and then **Access Reviews**, and then **Campaigns**.
3. For keyword search, type campaign name in the **Search** field and press Enter on your keyboard.
4. To view focused results, select one or more suggested filters available directly under the **Search** field.
5. To apply filters based on campaign status, use the drop-down menu in the top-right corner of the page. Select one from the following:
   - **My ongoing campaigns**: Displays campaigns with a **Status** of *In progress* or *Ready for approval*.
   - **My upcoming campaigns**: Displays campaigns with a **Status** of *Scheduled* or *Draft*.
   - **My previous campaigns**: Displays campaigns with a **Status** of *Approved*, *System ended* or *Terminated*.
   - **All campaigns**: Displays all available campaigns.

View Campaign Details

View campaign details to see the campaign description, selection criteria, review process, included identities or access control items considered for review, approval workflow, or perform additional actions on the campaign.

The information displayed and applicable actions vary based on the campaign state. Here's how you can view campaign details:

1. Go to the **Campaigns** page

2. For the campaign, select the corresponding **Menu** icon, and then select **View campaign details**.

Edit an Access Review Campaign

You can edit a *Draft* or *Scheduled* campaign before it is launched to modify the details. You can edit the selection criteria, workflow and reviewer details, campaign details, and schedule of campaign.

1. Go to the **Campaigns** page

2. For the campaign that you wish to edit, select the corresponding **Menu** icon, and then select **Edit**.

   The **Edit campaign** page provides the same guided workflow for entering your campaign parameters as the *Create* campaign page.

3. On the **Review and submit** step, select **Update**. Alternatively you can select **Back** to edit values, or **Cancel** to discard your changes.

Clone a Campaign

To generate similar access reviews, you can clone an existing campaign. The cloned campaign uses the existing access review criteria. However, you can modify the campaign details, campaign owner, or assign a new approval workflow.

Here's how you can clone an existing campaign:

1. Select one of the following options to clone a campaign:or from the **Actions** menu on the campaign detail page

   - On the **Campaigns** page, from the list, select **Menu** icon, and then select **Clone** corresponding to the campaign that you want to clone.
   - On the Campaigns detail page, from the **Actions** menu, select **Clone**

2. Select the **Clone** task to make a clone of the current access review campaign. You are taken to the **Clone campaign** page.
3. On the Clone campaign page, enter the following information
   a. **How often do you want this to run?** : Select **One time** to run a single occurrence of this campaign, or select a recurring pattern like **Quarterly**, **Monthly**, **Half-Yearly**, or **Yearly** to run this access review campaign periodically.
   b. **What do you want to call this campaign?**: Provide a name for the cloned campaign.
   c. **How do you want to describe this campaign?**: Provide a description for the cloned campaign.
   d. **Who owns this campaign?**: Provide details of the owner of the cloned campaign.

e. **How would you like to schedule your campaign?**: You can view this field only if you have selected to run your campaign one time. Select either **Run now** or **Schedule Later**. By default, the campaign is set to begin at the upcoming next hour, the following day of campaign creation.

f. **When do you want to Begin?**: If you have set a recurring pattern, then select the start date of when you want to begin the campaign series. By default, the campaign is set to begin at the top of the next hour, the following day of campaign creation. If you want to change this, select the **Select Date Time** icon and add a new date/time.

g. **When do you want to End?**: If you have set a recurring pattern, then select the end date of when you want to end the campaign series.

h. **Which approval workflow should be used?**: Select the approval workflow that you want to assign to this access review campaign. For details on how to create and manage approval workflows see Create Approval Workflow and Manage Approval Workflow.

4. When you have set your clone preferences, select **Create** to clone the current campaign.

   (Optional) To save your changes and later come back and edit the workflow or details, select **Save as draft**, or to cancel the current process, select **Cancel**.

Approve a Campaign

When the review tasks have been completed or the campaign due date has elapsed, the campaign moves to the *Ready for Approval* state. You can then *Approve* the campaign, and close the review process.

1. Go to the **Campaigns** page

2. For the campaign that you wish to approve, select the corresponding **Menu** icon, and then select **Approve**.

   You will see a confirmation message stating that the campaign has been approved. You can view the approved campaigns in the **my previous campaigns** or **All campaigns** category.

Terminate a Campaign

You can terminate a scheduled campaign at any point of time until a campaign is approved or system ended. For a periodic campaign, you can cancel all the ongoing (currently running) and upcoming (scheduled in future dates) access reviews campaigns for that series.

1. Select the **Terminate** task to terminate the current access review campaign.
2. On the **Confirmation** pop up dialog, select **Terminate** to end the campaign.
3. If you have a periodic campaign, and want to terminate entire series, select **Terminate Series**.

   For *Draft* campaigns, you can *Delete* the campaign.

View and Download Access Review Reports

Once the campaign is launched, you can view and download access review report for each campaign. In addition to viewing report, you can also save the reports offline in PDF format or download the CSV data for record-keeping or further analysis or audit.

You see a report displaying access review details and a breakdown of pending, approved, or revoked access review decisions for user role, user account and permission.

- For identity access reviews, you see the bifurcation of the review decisions based on top five organization, source organization, roles, and applications.
- For access control reviews, you see a breakdown of pending, approved, revoked, or modified access review decisions along with grouping of top five created since date ranges for identity collections or policies.

For example, if you have an identity review campaign setup that impacts applications and different roles, you should see the report displaying the top five applications and top five roles for which these access reviews are generated.

1. Go to the **Campaigns** page

2. For a campaign that you wish to view report, select the corresponding **Menu** icon, and then select **View report**.
3. If you want to retain a copy of the report, select **Download <certification type> PDF**, else select **Close** to return to the campaign.
4. To download CSV data:
   a. Select **Download CSV data** to generate a comma-separated values file with data for the campaign.
   b. On the **Download CSV data** confirmation pop-up dialog, select **Download** to download the CSV file.

Event-Based Micro Certifications

Micro-Certifications: Event Driven Access Reviews

Micro-certifications are automatically launched by Oracle Access Governance whenever an event, such as change event, timeline event, or unmatched account event, is detected. Oracle Access Governance continuously monitors identity profile and whenever a pre-defined event is detected, it launches access reviews related to that event. These generate near real-time access reviews so that prompt actions can be taken whenever these pre-defined events are detected. It also helps to reduce the certification fatigue as reviewers have to make a decision only for the affected identities.

As an *Administrator*, you can set up these micro-certifications from the **Access Reviews → Event-Based Setup** page. Enable or disable an event, auto approve low-risk items, auto-remove unmatched accounts from the system, or add an approval workflow. Reviewers can review or reassign the review tasks from the **My Access Review** page. Event-based access reviews have the **Event - <Identity Attribute Change>** identifier.

> **Note:**
>
> The **Event-Based Setup** menu option is not available when you have not activated any identities containing data for identity attributes. To view this option, you must activate at least one identity from the **Manage Identities** page. See Select Included Identities for details on how to enable identities in Oracle Access Governance.

Change Event

Change Events are triggered whenever changes are detected in an identity profile. Oracle Access Governance initiates real-time, focused access reviews based on occurrence of the events, such as job-code change, location change, manager change.

You can enable event-based access reviews for core attributes (for example, Job Code, Organization, Location, and so on) as well as custom attributes (for example, Cost Center, Project Code, and so on).

> **Note:**
>
> If you don't see the option for selecting custom attributes, contact the Oracle Access Governance Administrator. You first need to enable it from the **System Administration** settings within Oracle Access Governance Console. See View and Configure Identity Attributes.

Change Event is associated with joiner-mover-leaver (JML) actions.

- *Joiner* refers to action taken by the system when an identity joins the company, such as assigning some birth-right access privileges.
- *Mover* refers to action taken by the system when an identity moves within the same organization. For example, changes in access privileges due to internal job transfers or location change.
- *Leaver* refers to action taken by the system when an identity leaves the company, such as revoking access over all corporate applications and systems.
  Scenario: *Ema*, an employee at Acme corporation, has moved to a different project, reporting to a different manager within the same department. From an identity viewpoint, *Ema* no longer requires access privileges required by direct reports of her previous manager and project but now requires new access privileges.

  > **Note:**
  >
  > In this scenario, we are assuming *Manager* is the core attribute in your data schema and *Project Code* is one of the custom attributes in your data schema.

  For this, you need to enable event-based access reviews for the core attribute **Manager Change** and a custom attribute **Project Code**. Whenever the latest data synchronization happens from the orchestrated system with these updates, Oracle Access Governance automatically raise multiple event-based access reviews associated with this single identity.

Timeline Event

Timeline event access reviews are triggered annually on a given date. Oracle Access Governance automatically launches access reviews on the same day each year for that identity. This may refer to a specific event, for example an anniversary event such as an employee's organization joining date, or a software application license renewal date.

If configured, Access Reviews are generated on the specific date, to determine if permission associated with the event are still appropriate. Alternatively, you can configure a number of days prior to the event date on which to generate the review task.

Scenario: *Bill*, an employee at Acme corporation, uses the *CorporateLDAPdirectory* application. Bill's access to this application needs to be reviewed on an annual basis, based on the *ActiveStartDate* attribute. When Bill is first granted access to *CorporateLDAPdirectory* the *ActiveStartDate* is recorded. If you enable a timeline event on this application/attribute combination, then on the anniversary of Bill's first grant of the application, an access review will be generated, which allows a reviewer to revoke Bill's access to the application, or accept and allow Bill access to the application for another 12 months.

Unmatched Accounts Event

Unmatched Accounts events are triggered whenever Oracle Access Governance detects an orphan account, which cannot be associated with any identity.

You can select the orchestrated system for which you want to configure this event type. You can configure to auto-remove unmatched accounts.

Scenario: Oracle Access Governance detects orphan account of a former employee working at Acme corporation. It launches access reviews for this unmatched account event type. Reviewers can then opt to revoke accesses associated with this orphan account.

Configure and Manage Event-based Access Reviews

You can perform micro certifications in Oracle Access Governance using the *Event-Based Access Reviews*. You can configure one or more predefined event types, which when triggered, launches the access reviews automatically. You can choose to auto-approve the low-risk items, or reviewers can certify, that is accept or revoke accesses associated with the event.

Configure Change Event Access Review

Configure Change event access reviews from the **Event-Based Setup → Change** page to trigger automatic occurrence of access review whenever a change in identity profile is detected.

Here's how you can configure **Change** events:

1. Log on to the Oracle Access Governance Console with a user assigned the *Administrator* application role.
2. Select from the  navigation menu.
3. Click **Access Reviews** and then **Event-Based Setup**. The **Event-Based Setup** landing page is displayed.

4. On the **Event-Based Setup** page, select the **Change** tab. A list of available change events is displayed. Each change event has a status of **Enabled** or **Disabled** and an **Actions**

   drop-down menu , providing the option to **Edit** or **View details**.
5. Select **Edit** for the event-type you want to enable.
6. On the **Configure the event type** screen:
   a. To enable this event type, in the **Enable or disable this event-based access reviews** option, select **Enable**.
   b. If you want to auto-approve low risk task for this event type, select **Yes**.
7. Choose an approval workflow for this event type access review. A list of the available workflows is visible. For more details, see Create Approval Workflow and Manage Approval Workflow. Once you have selected your workflow, select the **View approval workflow** link to see a graphical representation of the selected workflow.
8. Select the scope of justification required for review decisions. You can select for reviewers to add comments for all the review decisions, for revoke decisions only, or keep the justification field as optional.
9. Select **Save**.

Configure Shared Workflow for Multiple Change Events

Shared Workflow or *Multi-event access review* is considered whenever multiple change events are triggered for a single identity within a short span of time.

. To configure the shared workflow:

1. Log on to the Oracle Access Governance Console with a user assigned the *Administrator* application role.
2. Select from the navigation menu. Click **Access Reviews** and then **Event-Based Setup**. The **Event-Based Setup** landing page is displayed.
3. On the **Change** tab, select **Edit shared workflow**.
4. On the **How do you want multi-event reviews to proceed?** screen:
   a. Confirm if you want to auto-approve low risk task for this event type by selecting **Yes** or **No**.
   b. Choose an approval workflow for this event type access review. A list of the available workflows is visible. For more details, see Create Approval Workflow and Manage Approval Workflow. Once you have selected your workflow, select the **View approval workflow** link to see a graphical representation of the selected workflow.
   c. Select the scope of justification required for access review decisions. You can select for reviewers to add comments for all the review decisions, for revoke decisions only, or keep the justification field as optional.
   d. Select the access review owner for this event type access review. By default, it is assigned to the administrator who configures this event type. Consider Fallback Mechanism in Access Review before adding an owner.
   e. Select **Save**.

Configure Timeline Event Access Reviews

Configure Timeline event access reviews from the **Event-Based Setup → Timeline** page to trigger automatic occurrence of access review annually on a particular date, such as job anniversary. By default, no automatic timeline event changes are per-configured. You must have at least one date attribute configured to enable this event type.

Here's how you can configure **Timeline** event:

1. Log on to the Oracle Access Governance Console with a user assigned the *Administrator* application role.
2. Select from the  navigation menu.
3. Click **Access Reviews** and then **Event-Based Setup**. The **Event-Based Setup** landing page is displayed.
4. On the **Event-Based Setup** page, select the **Timeline** tab.
5. To create events, select the **Create timeline event** button.
6. On the **Add a new timeline event configuration** screen:
    a. Select date attribute in the **Which date attribute should the event be triggered from?** list. Date attributes are identity attributes with a *Date* type, that are enabled for event-based campaigns. For further details on defining attributes, review View and Configure Identity Attributes.
    b. Enter number of days prior to the annual event date when the event should be triggered.
    c. Enter a unique event name in the **What do you want to name this event?** field.
    d. Choose to **Enable** or **Disable** the event-type.
    e. Select **Yes** to auto-approve low risk review task for this event type, else reviewers can take decision manually from the **My Access Reviews→ Identity** page.
7. Select the system for which you want to enable this event type. Based on your selection, a list of applicable applications are visible.
8. Select the applications you want to include in the timeline event change. By default all applications will be included in the review.
9. In the **Choose your Workflow** section,
    a. Choose an approval workflow for this event type access review. Consider Self Certification Guardrails and Fallback Mechanism in Access Review before configuring this workflow. For more details, see Create Approval Workflow and Manage Approval Workflow. Once you have selected your workflow, select the **View approval workflow** link to see a graphical representation of the selected workflow.
    b. Select the scope of justification required for review decisions. You can select for reviewers to add comments for all the review decisions, for revoke decisions only, or keep the justification field as optional.
    c. Select the access review owner for this event type access review. By default, it is assigned to the user who configures this event type. Consider Fallback Mechanism in Access Review before adding an owner.
10. Select **Save** to enable the event type changes.

Configure Unmatched Accounts Access Review

Configure unmatched account event access reviews from the **Event-Based Setup → Unmatched accounts** page to trigger automatic occurrence of access review whenever an orphan account is detected in Oracle Access Governance. Reviewers can review access for the unmatched accounts from the **My Access Reviews→ Ownership** page.

Create an Unmatched Accounts Event

To create unmatched account events, complete the following tasks:

1. Log on to the Oracle Access Governance Console with a user assigned the *Administrator* application role.
2. Select from the  navigation menu. Click **Access Reviews** and then **Event-Based Setup**. The **Event-Based Setup** landing page is displayed.
3. On the **Event-Based Setup** page, select the **Unmatched accounts** tab.
4. To create an unmatched account event configuration, select the **Create an unmatched accounts event** button. You are directed to the **Add a new unmatched account event configuration** page.

Configure an Unmatched Accounts Event

1. In **What do you want to name this event?**, add a meaningful name for the unmatched accounts event type.
2. To enable this event type, in the **Enable or disable this event-based access reviews** option, select **Enable**.
3. If you want the reviewer to take actions on the access reviews for unmatched accounts, in the **auto remove unmatched accounts**, select **No**.
4. To automatically remove all unmatched accounts reported by this event, in the **auto remove unmatched accounts** option, select **Yes**. All unmatched accounts will be removed from your environment including Oracle Access Governance and any Managed Systems from which the account was ingested.
5. Select one or more orchestrated systems for which you want to set up this event. By default, all the orchestrated systems are considered for the unmatched accounts event.
6. In the **Choose your Workflow** section,
   a. Choose an approval workflow for this event type access review. For more details, see Create Approval Workflow and Manage Approval Workflow. Once you have selected your workflow, select the **View approval workflow** link to see a graphical representation of the selected workflow. You can select
      - **Application Owner**, where access review owner will be assigned as the reviewer or certifier. Consider Self Certification Guardrails and fallback process before configuring this workflow.
      - **Custom User**, where any active identity available in Oracle Access Governance can be assigned as the reviewer.
   b. Select the scope of justification required for review decisions. You can select for reviewers to add comments for all the review decisions, for revoke decisions only, or keep the justification field as optional.

c. Select the access review owner for this event type access review. By default, it is assigned to the user who configures this event type. Consider fallback process before adding an owner.

7. Select **Save**.

View Event Details

As an *Administrator*, you can view details on each event-type configured for your application in the Oracle Access Governance Console. You can view the date when the event was enabled, selected rules and systems for the event-type, approval process details, along with campaign owner.

To view event-based settings:

1. Select from the navigation menu. Click **Access Reviews** and then **Event-Based Setup**. The **Event-Based Setup** landing page is displayed.
2. Select **View details** from the **Actions** drop-down menu for the event-type you want to view. The **Event - <event type name>** screen is displayed with the event details.

Edit Event-based Access Reviews

Update the existing event details, as follows:

1. Select the navigation menu.
2. Click **Access Reviews** and then **Event-Based Setup**. The **Event-Based Setup** landing page is displayed.
3. On the **Event-Based Setup** page, select the event type: **Change**, **Timeline**, or **Unmatched accounts** tab.
4. From the **Actions** menu, select **Edit**.
5. Update the details and select **Save**.

Delete Event Type for Access Reviews

As an *Administrator*, you can delete **Timeline** or **Unmatched Accounts** events. You can disable the **Change** event but cannot delete it.

Delete an existing event type, as follows:

1. Select from the navigation menu.
2. Click **Access Reviews** and then **Event-Based Setup**. The **Event-Based Setup** landing page is displayed.
3. On the **Event-Based Setup** page, select either **Timeline** or **Unmatched accounts** tab.
4. From the **Actions** menu, select **Delete**.

As an *Administrator* of Oracle Access Governance, you can analyze information on event-based access reviews by generating reports using the **Event-Based Report** capability of Oracle Access Governance.

Navigate to the Event-Based Access Reviews Report Service

You can run, view, and save the event-based access review report.

You must have *Administrator* rights to view and run this service. Find out more about application roles in the Understanding Application Roles topic

Here's how you can open the event-based access review report service on Oracle Access Governance Console:

1. From your browser, go to Oracle Access Governance. On the OCI console, you can open it using the **Service Home Page** link.
2. In the **Username** field, enter either your *Administrator* user name.
3. In the **Password** field, enter your password and select **Sign In**.

   You will be navigated to the home page of your Oracle Access Governance Console.
4. Click  icon on the top, left corner of the application page to display the navigation menu.
5. Select **Access Reviews**, and then **Event-Based Setup**

   The **Event-Based Setup** landing page is displayed.
6. Click the **View access review report** button

The **Event-Based Access Reviews Report** landing page is displayed.

Run Event-Based Access Reviews Report

As an Administrator, you can generate a monthly report on event-based access reviews by selecting appropriate reporting criteria.

You can generate the report based on date range and event types (predefined identity changes to perform access reviews). For example, you can generate a report on the access reviews initiated or implemented for all the new identities added or removed in your organization in the last month.

Here's how you can run an event-based access reviews report by selecting the appropriate report selection criteria:

1. In the **From** field, select the start date from which you want to run a report.
2. In the **To** field, select the end date up to which you want to run a report.

   To generate a report, you must take care of the following date range rules:

   - You can select only the first day of the month.
   - You can generate a report only for a maximum of a 12-month period, i.e. the difference between the From date and the To date cannot be more than 12 months.

   If you started implementing event-based access reviews from the 10th of this month and want to generate report till the present day, then in the **From** field, select the first day of

this month, and in the **To** field, select the first day of the next month. However, the results will include records only from the date of implementation of the event-based access reviews up to the current day.

3. Select the event type. Available options are:
   - **Change**
   - **Timeline**
   - **Unmatched accounts**
4. If you selected **Change** event type then select one or more predefined actions or triggered scenarios for your report. The available options are:
   - **Identity Enabled**
   - **Identity Disabled**
   - **Department**
   - **Job Code**
   - **Location**
   - **Manager**
   - **Source organization name**
   - **Multiple Event Changes**

   > **Note:**
   >
   > The **Enabled** and **Disabled** event-types displayed are based on whether the specific event is disabled or enabled in the **Event-Based Setup** screen.
5. If you selected **Timeline** or **Unmatched accounts** event type then select the Event Name you want to include in the report.
6. Select **Apply**.

As per the selected criteria, you can view the graphical charts on the same application page.

View Event-Based Access Reviews Report Results

After you generate an event-based access review report, you can see a set of graphical charts (donut charts, stacked bar charts, and chart legends) displaying the report details on the same application page.

The donut charts display the following information:

- **User accounts review decision**: Shows the number of impacted user accounts after running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.
- **Role membership review decision**: Shows the number of impacted user roles and group roles after running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.
- **Permission assignments review decision**: Shows the number of impacted user permissions or entitlements after running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.

- **Unmatched accounts review decision**: Shows the number of unnmatched accounts after running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.
- **Auto action of low risk**: Shows the number of approved low-risk review tasks after running the event-based access review. These low-risks review tasks are categorized based on **Auto**, which are automatically approved by Oracle Access Governance, or **Manual**, which are manually approved by the assigned reviewer.

The stacked bar charts display the following group information:

- **Group by user organization - Top 5**: Shows which top five organizations are affected after the running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.
- **Group by applications - Top 5**: Shows which top five applications are affected after the running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.
- **Group by Roles - Top 5**: Shows which top five roles are affected after the running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.
- **Group by system - Top 5**: Shows which top 5 integrations or orchestrated systems are affected after the running the event-based access reviews. The decision statuses are categorized based on *Pending*, *Approved* , or *Revoked* access reviews.
- **Group by insights**: Shows, for unmatched accounts, the distribution of reason for flagging as unmatched based on the following options:
  - **No match**
  - **Multi-match**
  - **User deleted**

Additional Actions

In addition to viewing report, you can also save the event-based access reviews report offline in the PDF format or download the CSV data for further analysis, audit, or anything else you usually do with offline reports.

Download PDF: Select the **Download PDF** button, which is available at the top, right corner of the application page, to save the report results offline in PDF format. Your PDF file will be saved to your downloads directory.

Download CSV Data: Select the **Download CSV Data** button, which is available at the top, right corner of the application page, to export the data in the CSV format.

Perform Access Reviews

Understanding Reviewer's Actions for Effective Access Certification

As an Access Reviewer, you can certify access privileges using the **My Access Reviews** feature. You can review identity, access control, and ownership review tasks, bulk approve low-risk items, check the AI/ML-equipped prescriptive analytic insights, review high-risk items, and make informed decisions based on AI/ML-driven recommendations provided by Oracle Access Governance. You can also reassign or delegate your review tasks to some other reviewer.

Access Review Task Types in Oracle Access Governance

After a campaign is launched, the **Campaigns** service in Oracle Access Governance actively tracks the access for the identities in scope, generates intelligent insights, and creates the access reviews. Based on the Access Review type, Oracle Access Governance generates identity review, access control, and/or ownership review tasks.

Here are a few details about each access review task:

Identity Review Tasks

Identity Review tasks include certification of identity access rights, evaluating user accounts, permissions, and roles. These are initiated when you launch an on-demand Identity Review task, or are initiated on occurrence of some identity event, such as department change, manager change, and so on. A single campaign may generate multiple review tasks. For example, when you launch *Identity Access Reviews*, Oracle Access Governance may generate access reviews tasks for accounts, permissions, or roles associated with a single identity on the **My Access Reviews → Identity** page.

Oracle Access Governance generates prescriptive insights and provide recommendations so that reviewers can make an informed decision to either **Approve** or **Reject** the required identity access.

Access Control Review Tasks

Access Control Review tasks include audit of Identity and Access Management (IAM) policies, and identity collections, initiated by on-demand *Policy Review* and *Identity Collections Review* campaigns. For example, when you launch review for domain administrator policy for your tenancy, Oracle Access Governance may generate access review tasks for the **Policy** type on the **My Access Reviews→ Access control** page.

Oracle Access Governance generates prescriptive insights and provide recommendations so that reviewers can make informed decision to either **Approve** or **Reject** entire policy at once, or make decision to **Approve** or **Reject** specific policy statement in that policy.

> **Note:**
>
> For an OCI policy, reviews are limited to the basic policy constructs. Advanced syntax including the "where" clause are beyond the scope of our supported feature.

Ownership Review Tasks

Ownership Review tasks include:

- Audit of Identity and Access Management (IAM) **Unmatched accounts**, initiated by event-based access reviews. These tasks help you to review any unmatched accounts for identities in Oracle Access Governance. While setting up an unmatched account event, you can select to remove the unmatched accounts automatically. As a reviewer, you can select an identity to match to, or you can remove the unmatched account
- **Ownership review** of Oracle Access Governance resources. These tasks help you to review and verify that only authorized owners are managing Oracle Access Governance resources. For example, you may want to run periodic campaigns to review group ownership of Identity Collections defined in Oracle Access Governance.
  You can view all the past ownership reviews run for the resource in the **Access Review trail** section. Based on the approval workflow selected in the campaign, either primary

owner or an active Oracle Access Governance workforce identity is chosen as the reviewer. As a reviewer, you can change primary and/or additional owners of the resources, certify the current ownership, or reassign the review task to some other active Oracle Access Governance user.

Intelligent Insights - Review Recommendations based on Prescriptive Analytics

Oracle Access Governance leverages prescriptive analytics to generate insights and recommend required actions on the review tasks. This enables access reviewers to make corrective decisions, lessen the administrative burden, and reduce cost.

Prescriptive analytics goes beyond prediction and involves action-oriented recommendations. These are data-driven guidance. Oracle Access Governance performs complex calculations and considers many dimensions such as organization, location, resource, and the sensitivity of that resource before recommending a decision. On a high-level, analysis of the permission is based on the following factors:

- Comparison with peers reporting to the same manager
- Comparison with peers with the same job code
- Comparison with peers in the same organization
- Recent changes in a user profile

As a reviewer, you get data-driven recommendation which simplifies the review process and mitigates the manual effort involved in identifying the anomalous permissions. From the **Insights** page, you can also track trail of reviews happened on a specific access, necessary for auditing purposes. You also get to track series of event changes involved for that access. All these details help you to make an informed decision for that access.

Audit Trail: Monitoring Access Review and Access Request Decisions

Audit Trail in Oracle Access Governance captures the history of request approval and access review tasks. It records decisions made during access requests and reviews, including whether access was accepted, rejected, or revoked, along with any justifications provided.

In Scope - What it Tracks

- Approval and access review decisions, including decisions to accept, reject, or revoke access, with justifications.
- Approve and revoke events for access granted by requests or directly assigned accesses in Managed Systems (Grant Type **Request** and Grant Type **Direct**).
- Revoke of identity accesses for event-based reviews that are triggered when there are identity attributes changes. This doesn't show access revoked through policy when there are attribute changes.
- Approval of access requests with SOD violations. These requests are identified by the  icon indicating that request was approved even though there were separation of duties violations from the Risk Management Cloud.

Out of Scope - What it Doesn't Track

- Access granted or revoked through policies, as these are reviewed at the policy level, not the identity level.
- Updates made directly in Orchestrated systems. For example, a role is removed from the Orchestrated Systems

Recent Change Events Log: Tracking Attribute Changes

The Recent Changes Log tracks identity attribute changes that are enabled in the **Event-based Setup**. For Campaigns (non-event-based), it shows all the change events for attributes with **Event-based Setup** enabled that occurred for the given identity over the past six months. For event-based campaigns, it logs only changes for the attribute triggering the review.

In Scope - What it Tracks

- Attribute change, such as department updates, if enabled in the Event-Based Setup.
- Changes in attribute that triggered the event-based reviews to perform micro-certification.
- For Campaigns, changes in identity attributes, enabled for Event-Based Setup, for the past six months.
- Access Request approval and reviews for accesses granted by requests or directly assigned accesses in Managed Systems.
- Revocation reviewed through event-based access reviews (not policy-based)

Out of Scope - What it Doesn't Track

- Specific access that was lost or gained due to attribute change.

Example: If a person's department changes and the department attribute has been enabled in **Event-based Setup** -> **Change events**, the changed attribute value will display in this section. It will only show which the changed attribute value and does not show what specific access was lost or gained due to the attribute change. For this mover scenario, all the accesses that the identity currently possesses need to be reviewed as a review task.

Delegating your Review Tasks

Delegating an access review task allows you to transfer your forthcoming review tasks to some other reviewers either temporarily or indefinitely. Typically, you would want to delegate a review item to some other reviewer or an identity collection during your absence, such as vacation.

With delegation, the ownership of review items does not change. A backup reviewer is assigned in absence of the intended reviewer so that no delays happen. On the **Insights** page, the reviewer can see complete details from the **Access Review Trail**. For example, as a manager going on vacation, you can delegate your review tasks to the team lead. During your absence, the team lead can continue to take decisions on your behalf which you can see from the **Access Review Trail**. However, the prime responsibility to review access review tasks will still be with the manager. You can delegate your access reviews using the self-service feature, that is from **My Stuff** → **My Preferences**. For more information, see Manage Delegation Preferences.

Reassigning a Review Task

Reassigning an access review task allows you to change reviewer for your pending review tasks to some other reviewers permanently. With reassignment, the ownership of review items changes. The review tasks are moved from the original reviewer and are assigned to the new reviewer. Only the new reviewer can see the reassignment details in the access review trail.

Typically, you would reassign your pending review items when there is a change in responsibility. For example, as a manager exiting the company, you can reassign your existing review tasks to your manager or your replacement. This shifts your pending review items to the new reviewer.

Bulk Changes - Managing Multiple Review Items Simultaneously

Oracle Access Governance allows you to approve, reject, or reassign multiple review items simultaneously, rather than making the same decisions individually. Reviewing multiple items at once reduces the administrative burden and saves time.

Use the data-driven recommendations to efficiently make a decision to approve or reject multiple requests at once. For example, while performing periodic access reviews for your team, you can approve all low-risk review items, with the **Accept** recommendation at once. You can even select multiple or all items to reassign the tasks to some other reviewer.

Here's what you can bulk-review for each review task:

- For identity review tasks, you can approve or reject multiple review tasks simultaneously. You can even reassign multiple identity review tasks at once.
- For access control tasks, for a policy, you can approve or reject all statements at once.
- For access control tasks, for an identity collection, you can approve or reject all members at once.
- For event-based reviews, you can configure to auto-approve low-risk tasks. You can also configure to auto remove unmatched accounts.

Bulk changes combined with prescriptive analytics allow you to speed up the process, improving operational efficiency without compromising security.

Perform Access Reviews - Evaluate and Certify Access Review Tasks

As an Access Reviewer, you can certify access privileges using the **My Access Reviews** feature. You can accept or revoke identity accesses, group memberships, policies, or unmatched accounts. These tasks can be carried out by any active user associated with a specific approval workflow.

For example, if *John* is the reviewer for *WorkflowA*, and *Susan* is reviewer for *WorkflowB*, then *John* can access review tasks associated with *WorkflowA*. *Susan* can access reviews associated with *WorkflowB*. If we have *Betty* who is not associated with any approval workflow, then that user cannot perform tasks against any reviews.

You can bulk approve low-risk items, check the AI/ML-equipped prescriptive analytic insights, review high-risks items, and make informed decisions based on AI/ML-driven recommendations provided by Oracle Access Governance.

Review Identity Access Tasks

Identity Review tasks include certification of identity access rights, evaluating user accounts, permissions, and roles. These review tasks can be generated by Campaigns, User-Created Access Reviews, or identity events. Reviewers can make decisions from the **My Access Reviews** page, considering recommendations driven by prescriptive analytics.

Search a specific access review task by task name, across the available tabs. You can apply the suggested filters to view focused results. View the count of total review tasks for each review type assigned to you. By default, you will see the **Identity review tasks** tab.

Here's how you can accept or revoke identity access review tasks:

1. In the Oracle Access Governance Console, select **Access Reviews**, and then **My Access Reviews** from the  navigation menu.

The **My Access Reviews** page is displayed.

2. Review the information listed for the assigned access review tasks. As a reviewer, you can see all the user-created, campaigns-initiated, and event-based review tasks.
3. Review the recommendations.
4. To view review insights, select the **View** link under **Insights** column, corresponding to each review tasks.
5. Choose one of the options to make a review decision:
   - On the **My Access Review** page, corresponding to each review task, select the row-level  tick icon to accept the access, or the  cross icon to revoke the access. Optionally, you may choose to reassign a review task.
   - On the **My Access Review** page, for each review task, select the check box, and then select the **Accept** button, or the **Revoke** button.
   - On the **Insights** page, select the **Accept** button, or the **Revoke** button.
6. In the confirmation pop-up dialogue, add comments or justification for your action, and then select **Submit**.

> **Note:**
>
> - If you revoke an **Account** task, then it will auto action to revoke all the related entitlement tasks.
> - If you accept the only entitlement (**Role** or **Permission**) for an account, then it will auto action to accept the related **Account** task.
> - When you revoke a review item, the item is remediated automatically. A request is sent back to the orchestrated system to revoke the item. No manual steps are required.

Review Policy and Identity Collection with Access Control Tasks

Access control review tasks include certification of Oracle Access Governance policies, identity collections, or *Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM)* policies or OCI groups. These review tasks can be generated by user-created, or on-demand policy review or identity collection campaigns. Reviewers can make decisions from the **My Access Reviews** page, considering recommendations driven by prescriptive analytics.

Search a specific access review task by task name, across the available tabs. You can apply the suggested filters to view focused results. View the count of total review tasks for each review type assigned to you.

On the **Insights** page, you can view our recommendation for the review task. On the left-panel, you can view the policy details. You can view the association details for the policy, and view a series of review tasks initiated for that policy or identity collection since the time it was granted.

To perform an access control review task:

Accept or Revoke Policy Review Tasks

1. In the Oracle Access Governance Console, select **Access Reviews**, and then **My Access Reviews** from the navigation menu. The **My Access Reviews** page is displayed.
2. Select the **Access Control** tab.
3. Review the information listed for the assigned review tasks. As a reviewer, you can see all the user-created, campaigns-initiated review tasks.
4. Look over the recommendations for each review task.
5. To view insights and make decision, select the **Actions** link under **Insights** column, corresponding to each review tasks. The **Insights** page is displayed.
6. Choose one of the options to make a review decision:
   - To review each policy statement individually, in the **Access association** section, select the row-level  tick icon to accept the statement, or the  cross icon to revoke the policy statement. Optionally, you may choose to reassign a review task.
   - To accept all policy associations at once, select **Accept all**, or else select **Revoke all**.

   > **Note:**
   >
   > The non actionable statements provide no access rights, therefore no action can be taken on those policy statements. For example, any rule statement that forms a construct which can further be used in other policy statements to provide access rights.

7. Select **Apply** to save your decision. In the confirmation pop-up dialogue, add comments or justification for your action, and then select **Submit**.

Accept or Revoke Identity Collection Review Tasks

1. In the Oracle Access Governance Console, select **Access Reviews**, and then **My Access Reviews** from the navigation menu. The **My Access Reviews** page is displayed.
2. Select the **Access Control** tab.
3. Review the information listed for the assigned review tasks. As a reviewer, you can see all the user-created, campaigns-initiated review tasks.
4. Look over the recommendations for each review task.
5. To view insights and make decision, select the **Actions** link under **Insights** column, corresponding to each review tasks. The **Insights** page is displayed
6. Choose one of the options to make a review decision:
   - To review membership for each identity within the identity collection, in the **Included named identities** section, select the row-level  tick icon to accept the

statement, or the  cross icon to revoke the policy statement. Optionally, you may choose to reassign a review task.

- To accept all memberships at once, select **Accept all**, or else select **Revoke all**.

> **Note:**
>
> If you choose to review OCI IAM group membership and it contains members whose assignments are managed by Oracle Access Governance, then you can accept or revoke only the directly assigned members. For members managed by Oracle Access Governance, you must create a separate campaign for OCI Access Bundles using the **Which permissions?** tile in the Oracle Cloud Infrastructure (OCI) system. For more information, see Review Access to Systems Managed by Oracle Cloud Infrastructure (OCI).

7. Select **Apply** to save your decision. In the confirmation pop-up dialogue, add comments or justification for your action, and then select **Submit**.

Review Unmatched Accounts with Ownership Tasks

Ownership review tasks include audit of unmatched accounts, initiated by event-based access reviews. These tasks help organizations review any unmatched accounts with identities in Oracle Access Governance. Reviewers can make decisions from the **My Access Reviews** → **Ownership** page, considering recommendations driven by prescriptive analytics.

Search a specific access review task by task name, across the available tabs. You can apply the suggested filters to view focused results. View the count of total review tasks for each review type assigned to you. On the **Insights** page, you can view our recommendation for the review task. On the left-panel, you can view the unmatched account information. On the right, you can view details, and make appropriate decisions based on them.

To perform an ownership review task:

1. In the Oracle Access Governance Console, select **My Access Reviews** from the  navigation menu. You navigate to the **My Access Reviews** page.
2. Select the **Ownership** tab.
3. Look over the recommendations for each review task.
4. To view insights and make decision, select the **Actions** link under **Insights** column, corresponding to each review tasks. The **Insights** page is displayed.
5. To associate an identity with the unmatched account, click **Select an identity** button.
   a. In the **Match account to identity** panel, select the desired identity either from the **Suggested identities** or the **All identities** tab.
   b. Select **Match**.
   c. Select **Apply** to save your decision. In the confirmation pop-up dialogue, add comments or justification for your action, and then select **Submit**.
6. To remove the unmatched account, select **Remove**. In the confirmation pop-up dialogue, add comments or justification for your action, and then select **Submit**. Optionally, you may choose to reassign a review task.

Review Resource Ownership with Ownership Task

Ownership review tasks include audit of ownership of Oracle Access Governance resources, initiated by ownership review campaigns from the Oracle Access Governance system. Reviewers can certify the current owners, change resource ownership, or reassign the review task to some other reviewer from the **My Access Reviews → Ownership** page.

Search a specific access review task by task name, across the available tabs. You can apply the suggested filters to view focused results. On the **Ownership** tab, you can apply filters using the **Assignment type** to view specific ownership task type. View the count of total review tasks for each review type assigned to you.

To perform an ownership review task:

1. In the Oracle Access Governance Console, select **My Access Reviews** from the  navigation menu. You navigate to the **My Access Reviews** page.
2. Select the **Ownership** tab.
3. Apply filters using the **Assignment type** to view ownership task.
4. To make decisions or change ownership, select the **View** link, corresponding to each review tasks.
5. To change ownership of the resource, select the **Change ownership** button.
    a. In the **Who is the primary owner** field, select the desired identity to whom you want to assign as the primary owner.
    b. In the **Who else owns it** field, select one or more additional owners for the resources. You can assign up to 20 additional owners.
    c. Select **Done**.

    In the **Ownership** section, you can view the updated owners in the list.
6. Optional: If you want to return to the original list of owners, select the **Reset changes** button.
7. Once confirmed:
    - To save and certify the updated list, select **Apply**.
    - To certify the original list of owners, select **Accept**.

    Optionally, you may choose to reassign a review task.
8. In the confirmation pop-up dialogue, add comments or justification for your action, and then select **Submit**.

As a resource owner, you can verify your changes by viewing the resource details, or by viewing the access details from the **My Access → Ownership** page.

Reassign a Review Task

Reassign a single or multiple review items to other reviewers. The review tasks are shifted from the original reviewer to the new reviewer. Only the new reviewer can see the reassignment details in the access review trail.

You cannot reassign self-review (one where a user is the beneficiary as well as the approver), delegated, or escalated review tasks. The **Reassign** button for these review tasks is disabled.

1. In the Oracle Access Governance Console, select **Access Reviews**, and then **My Access Reviews** from the navigation menu.

   The **My Access Reviews** page is displayed.

2. Choose one of the options to reassign a review task:

   - On the **My Access Review** page, for each review task, select the reassign icon to reassign the access.
   - On the **My Access Review** page, select the check box at the row-level, and then select the **Reassign** button.
   - On the **Insights** page, select the **Reassign** button.

3. In the **Confirmation** pop-up window:

   a. Select the reviewer to whom you want to reassign.

   b. Enter justification to reassign the review item.

   c. Select **Submit**.

      A confirmation message is displayed

8
Self Service

View Access Details and Manage Account

As an Oracle Access Governance user, you can view your own accesses from the **My Stuff**, and then **My Access** page. You can view comprehensive details on granted roles, permissions, accounts, ownership, organizations, identity collections, identity attributes, cloud resources, and policies. You can also change your account password if the account is provisioned within Oracle Access Governance.

Identities

While exploring your access profile details, you can view your associated roles, permissions, accounts, ownership, organizations, identity collections, identity attributes, cloud resources, and policies.

For Identities, you can see the following information:

Table 8-1 Identity Access Profile Information

Access Component	Description
Identity Collections	Count and details of the identity collection associated with the identity. This can either be Oracle Access Governance identity collection or an ingested identity collection, such as OCI groups.
Permissions	Count and access rights detail associated with this identity. It gives clarity of how this access was granted, for which resource this permission has been granted, and whether it is a role, permission, or a privilege assigned to the identity.
Organizations	Count and details of Oracle Access Governance organizations associated with the selected identity.
Accounts	Get count and account details associated with this identity. It gives you details like account name, the orchestrated system name associated with the account, resource name, how the access has been granted, password change status. When viewing your own accesses using the **My Access** menu option, if the account is provisioned within Oracle Access Governance and **Password Change status** flag is set to **Applicable** , then you can change your password. To do so, select **Change password** and follow the instructions to change your password.

Table 8-1 (Cont.) Identity Access Profile Information

Access Component	Description
Roles	Count and details of roles assigned to this identity using the Oracle Access Governance Access Control framework. If you want to see the ingested roles available from Managed Systems, then see the **Permissions** tab.
Policies	Count and details of policies used for granting access to the selected identity. You can further browse a policy to view policy statement details by selecting the **View details** link. The policies assigned can either be Oracle Access Governance policies or cloud policies ingested from OCI.
Cloud Resources	Count and cloud resource details that specify resource name, its type, the associated privilege granted to the identity along with the policy name that granted this privilege.
Ownership	Count and details of access controls components owned by this identity, such as identity collections, roles, policies,
Identity Attributes	Core and custom Identity attributes along with its value. The attributes are logically sectioned under meaningful headings for relevancy.

Change Account Password

You can change your account password from the Oracle Access Governance Console if your accounts are provisioned within Oracle Access Governance.

The **Grant Type** of the Permissions must either be *REQUEST* or *POLICY*. You cannot change password for the accounts that are directly provisioned. So, you can't change password for Oracle Identity Governance (OIG) and Oracle Cloud Infrastructure (OCI) accounts.

Password Change status flag should be other than *Not Applicable*, such as *Never Attempted* to change the password.

1. In the Oracle Access Governance Console, select the Navigation menu, and then go to **My Stuff→ My Access**.
2. On the **My Access** page, select the **Accounts** tab.
3. Select the **Change Password** button corresponding to the account for which you want to change the password.
4. Type your new password and re-enter to confirm your new password. Your new password must meet the criteria displayed on the page.
5. Click **Submit**.

You can manage your approvals using the Oracle Access Governance Console.

Manage Approvals

You can manage your approvals using the Oracle Access Governance Console.

1. In your browser, navigate to the Oracle Access Governance service home page.
2. On the Oracle Access Governance service home page, click on the  icon, then select **My Stuff** → **Approvals** to navigate to the **Approvals** page, which lists access requests requiring your attention. All requests requiring approval will be displayed. Requests are listed as one access per row. If a request is made for multiple accesses, for example access to a database, a directory, and a cloud service, then this will be displayed as 3 rows requiring separate approvals in your approval list.
3. Details of the access request displayed include the Requestor, the name of the Identity for whom the request is made, the Access, and Justification. Requests can be sorted by the following fields, in ascending or descending order:
   - **Respond By**
   - **Beneficiary**
   - **Requestor**
   - **Access**

   You can select requests using the checkboxes to the left, and either **Approve** or **Revoke** by clicking the corresponding buttons. You will be given the opportunity to add a justification, before confirming your decision.
   Alternatively, you can click on the view button for a specific request and view additional details such as **Access request trail** and **Access history**. You can make an approval decision in the view details page by selecting the **Actions** menu, and selecting **Approve** or **Reject**. You will be given the opportunity to add a justification, before confirming your decision. You can also request further information about the access request is required by selecting this option from the **Actions** menu.

   If you have access requests with segregation of duties violations, these will be flagged. You cannot directly approve these requests, you must select **View details** to display the details page for the request where you can review the violations before taking action.

Preventive Segregation of Duties (SOD) Analysis

Oracle Access Governance allows you to perform preventive segregation of duties (SOD) analysis for Oracle Fusion Cloud Applications orchestrated systems during the provisioning process through integration with . Segregation of duties (SOD) separates activities such as approving, recording, and processing tasks so an enterprise can more easily prevent or detect unintentional errors and willful fraud. SOD constrains duties across roles so that unethical, illegal, or damaging activities are less likely.

Segregation of Duties Analysis in Oracle Access Governance

When you configure an Oracle Fusion Cloud Applications orchestrated system you have the option to enable integration. is a security and audit solution that controls user access to your Oracle Cloud ERP financial data, monitors user activity, and makes it easier to meet compliance regulations through automation. One of the features of RCMS is the use of controls to analyze SOD analysis within the Oracle Fusion Cloud Applications orchestrated system.

To enable within Oracle Access Governance you should meet the following requirements:

1. Configure an Oracle Fusion Cloud Applications orchestrated system to manage permissions.
2. The Oracle Fusion Cloud Applications instance you are integrating with should have controls configured that define your SOD policies. provides a library of ready-to-use controls for high-risk business processes, such as, AP, AR, GL, Payroll, and Compensation. These controls can be updated to reflect your enterprise using the graphical workbench provided with RMC. For further information, refer to the documentation.

You can enable preventive SOD by configuring your Oracle Fusion Cloud Applications orchestrated system following the instructions in Integrate with Fusion Cloud Applications or Configure Orchestrated System Account Settings.

Once configured, Oracle Access Governance will use to check for SOD violations when a user makes an access request for an access bundle. When you make the request, a **Preventive SOD Analysis** activity is started, which can be monitored in the Activity Log. This activity will make a check against for any controls indicating that an SOD violation has taken place for the user and access requested. The **Preventive SOD Analysis** process runs asynchronously and returns results to the access request. The following rules apply to this process:

- Preventive SOD Analysis can only run against a user that has already been created in Oracle Fusion Cloud Applications and is available to the engine. Once this user is provisioned, any access requests made by the user will be analyzed by RMC if this option is enabled.
- Only one Preventive SOD Analysis task can run for a particular user at any one time. If your user creates a second access request while the Preventive SOD Analysis task from a previous access request is still running, then the second RMC request will fail. Other reasons why Preventive SOD Analysis task might fail include RMC unavailable, and no user account in Oracle Fusion Cloud Applications.
- Preventive SOD analysis is supported for requests for access bundles. Access requests for Oracle Access Governance roles are not supported for SOD analysis.

Example: Preventive Segregation of Duties in Oracle Access Governance

Let's look at an example of preventive segregation of duties in Oracle Access Governance in action. Consider the example where a user in your organization is promoted from AR Analyst to AR Manager. In order to carry out their new duties, the user requests access to the AR Manager access bundle in Oracle Access Governance.

When the access request is made, a Preventive SOD Analysis task is run for that user and RMC identifies some SOD violations which are flagged in the access request. An example of such a violation might be:

- The user's current permissions allow them *Create User* on Oracle Fusion Cloud Applications ERP, while the access bundle requested includes *Manage Compensation*.

This combination of permissions has a potential for payroll fraud by creating ghost employees and setting compensation. This conflict is flagged in the access request, so that the approver can review the information in the request, and log into RMC for further information if required. On this basis the approver can make an informed decision on whether to approve or reject the request, or to request further information from the person requesting the access.

View My Access Requests

You can view your access request in the Oracle Access Governance Console by following the steps below:

1. In your browser, navigate to the Oracle Access Governance service home page, and log in.
2. On the Oracle Access Governance service home page, click on the  icon, then select **My Stuff → My Access Requests**.
3. You navigate to the **My Access Requests** page, which displays a row for each access request requiring your input. You can display ongoing requests, previous requests, or all requests by toggling the value of the drop-down list in the top right hand corner of the page. Access requests can be for one of the following types:
   - **Pending approvals**
   - **Info requested**
   - **Preparing**: access requests display this state when preventive SOD analysis is in progress.
   - **Approved**
   - **Rejected**
   - **Deleted**
4. You can view the details of an access request by clicking the

   icon, and selecting **View details**. The **Details** page displays attributes of the request including:
   - **Assignment type**
   - **Requested on**
   - **Requested by**
   - **Status**
   - **LOB**
   - **Justification**

   The beneficiary of the access is available in the header of the detail page. The current state of the approval workflow is available in the View approval workflow panel.

   To view details of the approval request flow, select any of the rows in the **Access request trail** table. A diagram of the flow is displayed in a pop up, showing details of the step selected.

> **Note:**
>
> It is not possible to view the details of a request if it hasn't gone through an approval workflow process or it has been failed.

5. You can provide information for a request by clicking the icon, and selecting **Provide information**, or by selecting the **Provide information** button from the **View details** page. A pop-up is displayed, where you can enter the answer to any questions asked by an approver. Enter the details requested and click **Respond**. to complete.
6. If you no longer require a particular access request, you can cancel that request by clicking the icon, and selecting **Cancel request**.

As an Oracle Access Governance user you can request access to resources and roles. Requests can be made for yourself, or for others. This process creates an access request which is either granted without further action, or is subject to an approval workflow.

Request Access to a Resource

You can create a resource request using the Oracle Access Governance Console:

Request a new access (Resource)

To create a request for a particular resource:

1. In your browser, navigate to the Oracle Access Governance service home page, and log in as a user with the *Access Control Administrator* application role.
2. On the Oracle Access Governance service home page, click on the  icon, then select **My Stuff → Request a new access**. You navigate to the **Request access** page.
3. On the **Request access** page, select **Start** from the tile **Which access would you like?**. This takes you to the **Request a new access** workflow, where you can select resources such as applications, and cloud services and request access to them.
4. The first step of the workflow is **Select who**, which determines if the request you are creating is for yourself, or for another identity.
   a. If the request is for yourself, then select **Yes**, and click **Next**, to navigate to the next step of the workflow.
   b. If the request is for another identity, then select **No**, and select or search for the identities you want to request the resource for. Once you have selected your identities click **Next**, to navigate to the next step of the workflow.
5. On the **Select access** page, select or search for the resources you want to get access to, and click **Next**.

6. The next step of the workflow is **Add details**. The details requested will depend on the resource selected, but will typically be something like a justification for granting the access requested. Enter any details required and click **Next**.
7. From the **Review and submit** page, select **Submit request** if you are happy with your selections.

Request Access To A Role

You can create a role request using the Oracle Access Governance Console:

Request a new access (Role)

To create a request for a role:

1. In your browser, navigate to the Oracle Access Governance service home page, and log in as a user with the *Access Control Administrator* application role.
2. On the Oracle Access Governance service home page, click on the  icon, then select **My Stuff → Request a new access**. You navigate to the **Request access** page.
3. On the **Request access** page, select **Start** from the tile **Which role would you like?**. This takes you to the **Request a new role** workflow, where you can select roles defined within your environment, and request that you be assigned these roles.
4. The first step of the workflow is **Select who**, which determines if the request you are creating is for yourself, or for another identity.
   a. If the request is for yourself, then select **Yes**, and click **Next**, to navigate to the next step of the workflow.
   b. If the request is for another identity, then select **No**, and and select or search for the identities you want to request the resource for. Once you have selected your identities click **Next**, to navigate to the next step of the workflow.
5. On the **Select access** page, select or search for the roles you want to assign, and click **Next**.
6. The next step of the workflow is **Add details**. The details requested will depend on the resource selected, but will typically be something like a justification for granting the role requested. Enter any details required and click **Next**.
7. From the **Review and submit** page, select **Submit request** if you are happy with your selections.

Delegation preferences allow you to setup and manage a delegate for certain tasks and activities in Oracle Access Governance in the event that you are unable to carry out the task, for example due to absence from the office.

Users can delegate tasks/activities using the Oracle Access Governance Console. You can use the **My Preferences** page or the Identity details page to assign delegated tasks/activities to another user or identity collection. You can choose when to start the delegation process and also specify the duration of the delegation. In Oracle Access Governance, you can delegate:

- **Access Reviews**: Some other user or identity collection can perform access reviews on your behalf
- **Approvals**: Some other user or identity collection can perform approvals on your behalf

You may want to delegate approvals or access reviews for the following reasons:

- Unavailability because of vacation, sickness, or working on other tasks
- To have the most qualified person to make the decisions
- Develop someone else's ability to handle additional assignments

Delegation settings can be accessed from a number of paths in the Oracle Access Governance Console:

- **My Stuff → My Preferences**: This option takes you directly to the **Delegation** tab, where a user can setup delegations for tasks assigned to themselves.
- **My Stuff → My Access**: From the identity detail page you can select the **Delegation** tab and setup your own user's delegation preferences.
- **Who Has Access To What → My Direct's Access**: A user's manager can select users they directly manage from this page, and amend delegation details for the selected user.
- **Service Administration → Manage Identities**: An administrator can select **View Details** or **Manage delegations** for a user listed in the **Manage Identities** page, and amend delegation details for the selected user.

The user types that can create and manage delegations are:

- **Administrator**: Users with the `AG_Administrator` application role can update their own delegations, and delegations for other users.
- **Managers**: Users that manage a user can update their own delegations, and delegations for users they manage directly,
- **Users**: Users can maintain their own delegation preferences.

Set up Your Delegation Preferences

Perform the following steps to navigate to the **Delegations** tab:

1. Log in to the Oracle Access Governance Console.
2. Navigate to the delegations tab using one of the following paths:
   - **My Stuff → My Preferences**.
   - **My Stuff → My Access → Delegations**.
   - **Who Has Access To What → My Direct's Access → <User> → Delegations**.
   - **Service Administration -> Manage Identities → <User> → View Details/Manage delegations → Delegations**.

Perform the following steps to add a new delegation:

1. In the Delegations screen, click the **Add a delegation** button.
   You will be navigated to the Add a delegation pop-up window.
2. Select what you want to delegate in the **Which tasks do you want to delegate?** field. You can delegate the following tasks:
   - **Access reviews**
   - **Approvals**
3. The field **Who do you want to delegate to?** allows you to either delegate the selected task to an individual or an identity collection.
   - If **An individual** option is selected, enter the name of the delegator in the **Who?** field

- If **An identity collection** option is selected, enter the name of the identity collection group in the **Who?** field

> **Note:**
>
> The Identity collection can have one or more than one member in it.

4. Select the date range for the delegation from the **How long do you want the delegation to last?** field. It can be either an indefinite time or a specific time range. Selecting:
   - **Indefinitely**: Allows you to set the delegation for an indefinite time.
   - **During a time range**: Allows you to select a date range for the delegation.
5. Select whether to send notifications to the original assignee. You can chose to send notifications to the delegator and the delegate, or to the delegate only using this check box. When setting a delegation preference for yourself, this checkbox is selected by default. For delegation preferences set by others, the checkbox is not selected by default. Set the **Include original assignee in notifications** checkbox as required.
6. Click **Save**.

Tasks or activities created after assigning a delegate are visible on both the dashboards of the delegator (the one who delegated the task) and the delegate (the one to whom the task/activity was delegated to).

> **Note:**
>
> Tasks or activities that were created before delegation will not appear on the delegate's dashboard immediately. It may be a few hours before the existing tasks/ activities are processed and displayed.

Edit a Delegation

Perform the following steps to edit a delegation:

1. In the Delegations screen, click the **Edit** button.
   You will be navigated to the Edit pop-up window.
2. You can change the individual assignee or identity collection to which you have delegated the task, by updating the **Who?** field. You cannot change assignee from an individual to an identity collection, or identity collection to an individual.
3. Select the date range for the delegation from the **How long do you want the delegation to last?** field. It can be either an indefinite time or a specific time range. Selecting:
   - **Indefinitely**: Allows you to set the delegation for an indefinite time.
   - **During a time range**: Allows you to select a date range for the delegation.

> **Note:**
>
> You can only modify the dates, not the **delegate** type.

4. Click **Save**.

Delete a Delegation

Perform the following steps to delete a delegation:

1. In the Delegations screen, click the  button.
2. In the Confirmation pop up, select **Delete** to remove the delegation or **Cancel** to retain the delegation.

9

Data Feed

Event Data Publisher in Oracle Access Governance

Event Data Publishing is a process to export one-time and sequentially and continually publish ongoing data events to external systems, such as an Oracle Cloud Infrastructure (OCI) cloud account. With Oracle Access Governance, you have the flexibility to export one-off and continually publish data events, such as identity, identity collections, policy, resource, access to resources, and so on, to your cloud tenancy. You may use this data for deriving insights, storing data for compliance, or for analyzing access management and governance data.

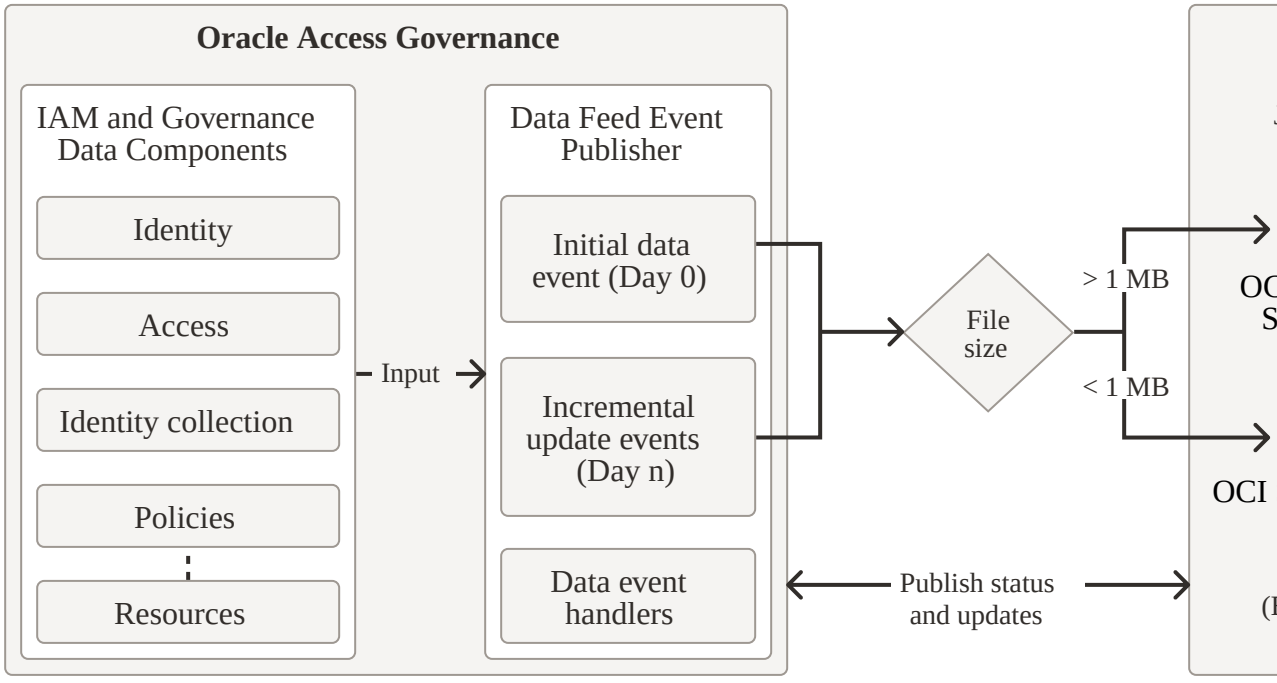
An event refers to any change in data state that occurs when there's creation, modification, or deletion of Oracle Access Governance components, such as identity, policies, resources, and so on.

Using Event Data Publisher, administrators get the complete control over access and identity data and may use it to automate event logging, streamline compliance reporting. For example, you may enable Data Feed to view insights on access violations in real-time.

Understanding Data Event Publishing Flow

Let's understand the Data Event Publishing flow in Oracle Access Governance:

Data Event Publishing flow uses OCI Buckets for one-off export, and publish subsequent updates either to OCI Streams or OCI Buckets depending on the file size.



1. Perform preliminary steps within your OCI cloud account to receive the data. For details, see Set Up OCI Tenancy for Data Publisher and Streaming.
2. Use the **Data Feed** service of Oracle Access Governance to publish the data to your cloud system.
3. Establish a connection with Oracle Access Governance by entering the configuration details. For details, see Configure Event Data Publisher in Oracle Access Governance.
4. The **Data Feed** event publisher service extracts the data components available in Oracle Access Governance. On Day 0, the initial event, depending on the file size, it captures and exports a complete snapshot of data components either to OCI Object Storage Bucket as JSONL files or as stream message to the OCI Streams. Mostly, Day 0 events will be published to OCI Object Storage Bucket.
5. The Data Event Handler maintains the publishing status and returns a success or failure notification.
6. For **Day N**, whenever there's a change in the state of data components, related to creation, modification, or deletion of components, the **Data Feed** event publisher detects the change in real-time, and publishes the sequential messages to the OCI Streams. The maximum size of a message can be 1 MB.
7. For **Day N**, if the message size is greater than 1 MB, the **Data Feed** event publisher publishes the updates to OCI Buckets as a new version appending or replacing entries in the versioned JSONL file.

Initial Data Event and Incremental Data Events : Day 0 and Day N Events

You can export and publish Oracle Access Governance data to either to OCI Streams or OCI Buckets.

Day 0 Event Publishing with Buckets

On Day 0, which is the initial data export, if the file size if greater than one megabyte (`> 1MB`), a complete snapshot for the supported data components in Oracle Access Governance is exported to the OCI Bucket as JSONL files (`.jsonl`). These files can handle multiple JSON objects in a single payload efficiently. You'll receive multiple files per data component. If the file size is less than one megabyte, then you'll receive OCI Stream messages.

On Day 0, the **Data Feed** event publisher sends the start message to OCI Bucket with messageType as `Day0`, operation `CREATE` and status `START`. Until all the data objects are exported, each of the Day 0 output object file contains the status `In_PROGRESS`. Once completed, the **Data Feed** event publisher sends the end message for Day 0. with the status either `SUCCESS` or `FAILED`. The success or failure status is displayed on the Oracle Access Governance Console, including the name of the administrator who performed the publishing operation.

Day N Event Publishing with OCI Streams or OCI Buckets

After Day 0, subsequent updates are published in real-time either to OCI Streams or to OCI Buckets. Each message published onto the streams has an `eventTime` attribute to help the consumer service manage the ordering of events.

Depending on the file size, the publishing destination is determined.

- If the file size is less than one megabyte (`< 1 MB`), then updates are published to OCI Streams in the JSON format. The data in streams is base64 encoded, ensuring fast transmission. To consume the message, you need to decode the data, and then leverage it for further use.

- If the file size is greater than one megabyte (`> 1 MB`), then updates are published to OCI Buckets as versions to Day 0 output files, in the JSONL format.

Available Data Components for Publishing

You can export and publish the following data components:

Identity

Scope: All Active Identities (Workforce or Consumers)

messageType: `IDENTITY`

All the active identities, workforce or consumers, are published as `IDENTITY` events to OCI Buckets and OCI streams. The output file contains the composite identity profile details, tagged as the `globalIdentity` attribute. It contains access profile details, including core and custom attributes. A composite identity profile in Oracle Access Governance is built up using attributes from one or more orchestrated systems. For example, `jobCode` of an identity may be ingested from PeopleSoft and identity profile details, such as firstName, lastName from Oracle Fusion Cloud Applications. Oracle Access Governance uses attributes listed in `globalIdentity` as a source of truth to perform various governance and provisioning operations.

Additionally, it contains an array of identity attributes incoming from other integrated orchestrated systems, tagged as `targetIdentities` matched with this composite identity profile . For example, a composite identity profile in Oracle Access Governance uses identity details from Oracle Fusion Cloud Applications, but if the same identity is available in PeopleSoft and matched with the composite identity, then the identity attributes available in PeopleSoft will be published as part of `targetIdentities`. For schema reference and attribute details, see Identity Reference Schema and Sample.

Identity collection

Scope: OCI IAM Groups

messageType: `GROUP`

All the available OCI IAM group ingested into Oracle Access Governance will be published as `GROUP` events. The output file contains OCI identifiers, such as domain id, compartment id, identity collection name, description, identity collections with an array of identities included in an identity collection. *Update* and *Create* operations share the same schema. However, when a new identity collection is created, you'll not receive any identities in remove attribute. For schema reference and attribute details, see Identity Collection Reference Schema and Samples.

Policy

Scope: OCI Policies

messageType: `TARGET_ACCESS_POLICY_STATEMENT`

All the available OCI policies ingested into Oracle Access Governance will be published as `TARGET_ACCESS_POLICY_STATEMENT` events. For an OCI policy, the output file contains OCI identifiers, such as domain id, compartment id, policy name, description. It contains policy details, including the subject to whom the access is granted, access type and scope of access granted. For schema reference and attribute details, see Policies Reference Schema and Sample.

Resource

Scope: All resources

messageType: `RESOURCE`

All available resources across all orchestrated systems ingested into Oracle Access Governance will be published as `RESOURCE` events. The output file contains resource identifiers within Oracle Access Governance and in OCI, resource name, description, and resource type. For schema reference and attribute details, see Resource Reference Schema and Sample.

Access

Scope: OCI policies and OCI resources

messageType: For policy to resource Mapping, it is `POLICY_STATEMENT_RESOURCE_MAPPING`. For resource to policy mapping, it is `RESOURCE_POLICY_STATEMENT_MAPPING`.

Access component contains two methods to see the same data:

- **Policy Granting Access to Resources**: List of resources governed by a specific policy. Each JSON object contains a policy statement detailing a set of resources attached to it. In this data, the focus is on a specific policy governing a set of resources.
- **Resources Access using Policy Statements**: List of policies associated with a resource. Each JSON object contains a resource detailing set of policies applied to it. It is the reverse process and focus is on a specific resource.

For schema reference and attribute details, see Access Policy to Resource Schema and Sample and Access Resource to Policy Schema and Sample.

Configure Event Data Publisher in Oracle Access Governance

You can publish the data events from Oracle Access Governance to your OCI tenancy using the **Data Feed** feature in the Oracle Access Governance Console. You first need to perform some preliminary configurations in your tenancy, and then add the connection details in the Oracle Access Governance Console. Once the connection details are validated, the first data event is published in your Object Storage Bucket, and all the subsequent updates are received continually and sequentially either in OCI Bucket or OCI Streams.

Prerequisites

You can seamlessly export Oracle Access Governance data into your OCI tenancy. Here are a few mandatory requirements to consider before you proceed with the set up.

- You must be assigned an Oracle Access Governance Administrator role `AG_Administrator`
- You must have an active Oracle Cloud Infrastructure (OCI) orchestrated system integrated with Oracle Access Governance to view the **Data Feed** menu under **Service Administration**.
- Oracle Access Governance service instance, service account, object storage, and other related resources must all reside in the same region and identity domain.
- Your Oracle Access Governance service instance tenancy can be different to the tenancy where you want to receive the data as long as they are located in the same region.

- If the region of your OCI resources is different than the Oracle Access Governance service instance region, then replicate the identity domain in the Oracle Access Governance service instance region. For more details, see Replicating an Identity Domain to Multiple Regions.
- You may connect multiple Oracle Access Governance service instances available in different tenancies within a region to the same OCI resources. This means that data available in your multiple Oracle Access Governance service instances can be collected within the same bucket and stream. You may also choose to create separate OCI resources per Oracle Access Governance service instance.
- Ensure that your cloud account, including associated Object Storage buckets and Streaming services, has sufficient space and capacity before exporting object types. Review your account's Object Storage Quotas and Limits on Streaming Resources to avoid disruptions during the process.

Set Up OCI Tenancy for Data Event Publisher

Before you can use the **Data Feed** feature in Oracle Access Governance to publish your data, you must create a few OCI resources to support this. You need to create a compartment, a service account, an IAM group, generate API Keys and Authentication Token for the service account, Create Buckets, OCI Streams, and assign appropriate policies to give the group and service accounts access to related resources.

Step 1: Create a Compartment

Create a compartment dedicated for data feed so that you can assign restricted policies to access or modify resources in this compartment. You may skip the step if you already have a compartment. This compartment must contain all the required IAM resources (Domain, Groups, User), Object Storage (Buckets), and Streams for publishing event data.

1. Sign in to the Oracle Cloud Infrastructure Console as a tenancy administrator.
2. Open the navigation menu and select **Identity & Security**.
3. In the Identity section, select **Compartments**.
4. Select **Create Compartment** and add compartment name and description.
5. Confirm and select the **Create Compartment** button.

`data-feed-compartment`

For more details, see To create a compartment.

Step 2: Create a New Domain

Create a new domain dedicated for data feed in the compartment created in Step 1.

1. In your OCI cloud account, open the navigation menu and select **Identity & Security**.
2. In the **Identity** section, select **Domains**.
3. Select the compartment
4. Select **Create domain** and enter the basic details, such as name and description.
5. On the left pane, in the **Compartment** list, select the compartment created in the Step 1.
6. You may choose to select the default domain or create a new domain.

For more details, see Create Identity Domain.

`domain-feed-domain`

Step 3: Create IAM Group

1. On the OCI Console, open the navigation menu and select **Identity & Security** -> **Domains**.
2. Choose the domain created for data feed operations.
3. On the left pane, select **Groups**. A list of the groups available in your domain is displayed.
4. Enter the following details:
   - **Name**: Enter group name.
   - **Description**: Enter some descriptive information about the group.
5. Select the **user can request access** check box.
6. Select **Create**.

```
writer_access_group
```

Step 4: Create a Service Account by adding an Identity User

Create a new identity user for service account purposes. This user is provided restricted access through policies and is only used for programmatic access to OCI resources by Oracle Access Governance . Assign this service user to the IAM Group created in Step 3.

1. On the OCI Console, open the navigation menu and select **Identity & Security** -> **Domains**.
2. Choose the domain created for data feed operations.
3. On the left pane, select **Users**. A list of the users available in your domain is displayed.
4. Enter the first name, last name, username or email address.
5. Choose the IAM group to assign the user to the identity group.
6. Select **Create**.

Step 5: Create API Key

Create API keys to establish secure authentication between user cloud account and Oracle Access Governance service. Using this service account, it enables the service to perform appropriate operations on the OCI resources .

1. In the service account user page, on the left pane, in the **Resources** section, select **API keys**.
2. Select the **Add API key** button, and then select **Generate API key pair**.
3. Download the public key and private key.
4. Select **Add**. The configuration file is created displaying *ocid*, *fingerprint*, *tenancy* and *region* details.
5. Open the private key file (.pem extension) with any text editor, and save the information available on the file. You'll need this to configure the data publisher feature in Oracle Access Governance.

Step 6: Generate Authentication Token

Auth Tokens are Oracle-generated authentication tokens to authenticate and authorize the service account user to interact with OCI resources.

1. In the service account user page, on the left pane, in the **Resources** section, select **Auth Tokens**.
2. Click the **Generate token** button.
3. In the **Generate token** window, enter a meaningful description and then click **Generate token**.

The token is generated. Copy the displayed token and save it to a secure location. You'll need this to configure the data publisher event feature in Oracle Access Governance.

Step 7: Create a New Stream

Oracle Access Governance will publish the data sets smaller than `1 MB` to your OCI Streams. Generally, all the Day N events, having small-size updates, will be published to the streaming service set up in your tenancy.

1. On the OCI Console, open the navigation menu and select **Analytics & AI**.
2. Under the **Messaging** section, choose **Streaming**.
3. Enter a unique stream name.
4. Choose the data publisher specific compartment.
5. For stream pool, choose either to automatically create a default stream pool or to create your own stream pool.
6. In the **Define Stream Settings** section:
   a. In the **Retention (in hours)** field, enter a number from 24 to 168 for retaining messages in this stream. You can choose to enter any number. However, we recommend to enter the maximum number, which is 168 hours.
   b. Enter the number of partitions you want to create in your stream. The maximum limit depends on your tenancy limits.
7. Click **Create**. Your new messaging stream is created.

For more information, see Creating a Stream.

Save Streaming Details

Once created, select the stream name link to view its details. To configure your stream in Oracle Access Governance, copy and save the following details related to your stream:

- Stream Name
- Message Endpoint
- Stream OCID

For Stream Pool, follow the steps to view and save the information:

1. On the Stream details page, select the Stream Pool link.
   - Copy and save the Stream Pool OCID.
   - For viewing the **Bootstrap Server** link:
     a. On the left pane, in the **Resources** section, select **Kafka Connection Settings**.
     b. In the **Kafka Connection Settings** section, copy and save the **Bootstrap Servers** link.

Step 8: Create a New Object Storage Bucket

Oracle Access Governance will publish the initial data event to the Object Storage Bucket. Regular updates with data larger than `1 MB` will be published to Buckets.

1. On the OCI Console, open the navigation menu and select **Storage**.
2. Under the **Object Storage & Archive Storage** section, choose **Buckets**.
3. Choose the data publisher specific compartment and then click **Create Bucket**.
4. In the **Create Bucket** window, enter the bucket details.
5. Select the **Enable Object Versioning** checkbox.
6. Click **Create**. Your new bucket stream is created.

For more information, see Creating an Object Storage Bucket. Save the bucket name and namespace. You'll need these to configure the data event publisher feature in Oracle Access Governance.

```
Data-event-publisher-bucket
```

Step 9: Set Up Policies for Data Event Publisher

You'll need to set up IAM policies to enable the Oracle Access Governance Data Event Publisher to access the OCI resources.

1. On the OCI Console, open the navigation menu and select **Identity & Security**, then **Policies**.
2. Under **Identity**, select **Policies**.
3. Enter name and meaningful description.
4. In the **Policy Editor** section, use the toggle button to switch to manual editor.
5. Enter the policy statements as follows:

```
Allow group '<domain-name>'/'<group-name>' to use stream-push in
compartment <compartment-name> where target.stream.id
          = '<stream-OCID>'
```

```
Allow group '<domain-name>'/'<group-name>' to manage object-family in
compartment <compartment-name> where target.bucket.name
          = '<bucket-name>'
```

For example

```
Allow group 'data-feed-domain'/'writer_access_group' to use stream-push in
compartment data-feed-compartment where target.stream.id
          = 'ocid1.stream.oc1.iad.amaaaaaa1212121212126pc5dc6wjn7xloxga'
```

```
Allow group 'data-feed-domain'/'writer_access_group' to manage object-
family in compartment data-feed-compartment where target.bucket.name
          = 'Data-event-publisher-bucket'
```

Configure Settings for Data Publisher in Oracle Access Governance

You can configure settings in the Oracle Access Governance Console to start receiving the data events in your OCI tenancy. Use the **Data Feed** functionality to configure data publishing events.

You must complete the preliminary set up in your OCI tenancy before configuring the data feed functionality in Oracle Access Governance.

Navigate to the Data Feed Page

1. On the Oracle Access Governance Console, select the navigation menu icon and then select **Service Administration**.
2. Select **Data Feed**.

Enable the Data Feed Action

1. On the **Data Feed** page, from the **Actions** menu, select **Manage settings**.
2. Select **Yes** in the **Do you want to enable the data feed?** option. You'll see the configuration fields to enter details.

Add OCI Object Storage Bucket Details

Enter your OCI bucket details. You'll get the required information from the bucket details created in your OCI tenancy. For more information, see Step 5: Create API Keys and Step 8: Create a New Object Storage Bucket.

1. In the **Which region is the bucket in?** field, enter the region identifier where you have created the bucket. For example, for Ashburn region, enter `us-ashburn-1`. You can view the region name in the top navigation menu of your OCI Console and corresponding region identifier from Regions and Availability Domains.
2. In the **Which namespace is the bucket in?** field, enter the bucket namespace. For more information, see Getting an Object Storage Bucket's Details.
3. In the **What is the bucket name?** field, enter the bucket name.
4. In the **What is the tenancy OCID?** field, enter the tenancy OCID. You can view the tenancy details in your cloud account profile or from the configuration file. For more information, see Getting My Profile Details.
5. In the **Which region is the user in?** field, enter the service account user region identifier. You'll get the details in the configuration file. For more information, see Create API Keys.
6. In the **What is the user's OCID?** field, enter the OCID of the service account. You'll get the details in the configuration file. For more information, see Create API Keys.
7. In the **What is the user's fingerprint?** field, enter fingerprint value. You'll get the details in the configuration file. For more information, see Create API Keys.
8. In the **What is the user's private SSH key?** field, copy and paste the PEM file contents, starting from `-----BEGIN PRIVATE KEY-----` to `-----END PRIVATE KEY-----`. You should have this downloaded while creating the API keys for the service account. For more information, see Create API Keys.

Add OCI Stream and Stream Pool Details

1. In the **What is the auth token?** field, enter the user authentication token value. For more information, see Step 6: Generate Authentication Token.

2. In the **Which tenancy?** field, enter the tenancy name. You can view the tenancy details in your cloud account profile. For more information, see Getting My Profile Details.
3. In the **What is the username?** field, enter the service account user name prefixed with domain name in the format `<domain-name/user name>`. For example, if the domain name is *data-pub* and service account user name as *john.doe*, then enter `data-pub/john.doe`.
4. In the **What is the message endpoint?** field, enter message endpoint without `https` or bootstrap server value. For more information, see Streaming Details.
5. In the **What is the topic name?** field, enter stream name. For more information, see Streaming Details.
6. In the **What is the stream pool OCID?** field, enter the stream pool OCID. For more information, see Streaming Details.

> **Tip:**
>
> Enter Stream Pool OCID and not Stream OCID.

7. Click **Save**. The configuration details are saved. If there are any validation errors, you must resolve those before you could save the details.
8. Under **Actions**, click **Publish data now**. A confirmation message displays and then click **Publish**.

Once you publish the data, the publishing status along with date and time is displayed on the Oracle Access Governance Console. Depending on the data size, it may take a couple of minutes to publish the entire data. Once completed, you will see the completion status on the Console with a message, "A data publish is completed successfully on <Date> <Time> by <Administrator Email Address>." In the OCI bucket, you should see multiple event files under the `json` folder.

Event Data Publishing Reference Schema and Sample Files

Defines schema and sample output code snippet of Oracle Access Governance components published to Oracle Cloud Infrastructure (OCI) Buckets and OCI Streams.

Header Schema and Sample Output Reference

There are headers related to event types, covering **Day 0** and **Day N** export, and another event types, covering for publishing of data objects, which includes policies, identities, resources, and so on for create, update, and delete operations.

Day 0 Message Header Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "object",
  "properties": {
    "eventId": {
      "type": "string"
    },
    "correlationId": {
      "type": "string"
    },
```

```
    "eventTime": {
      "type": "string"
    },
    "eventTypeVersion": {
      "type": "string"
    },
    "version": {
      "type": "string"
    },
    "operation": {
      "type": "string"
    },
    "messageType": {
      "type": "string"
    },
    "eventType": {
      "type": "string"
    },
    "opcRequestId": {
      "type": "string"
    },
    "tenancyId": {
      "type": "string"
    },
    "serviceInstanceId": {
      "type": "string"
    }
  },
  "additionalProperties": false
}
```

Day 0 Sample Header

```
{
  "headers": {
    "eventId": "752d5e14-a784-4d91-9cf4-57c0a72d7620",
    "correlationId": "9a0041f5-f67f-4b06-8fbd-c9b64d1d5ee3",
    "eventTime": "2024-09-05T16:57:59.922065942Z",
    "eventTypeVersion": "1.0",
    "version": "1.0",
    "operation": "CREATE",
    "messageType": "DAY0",
    "eventType": "com.oracle.idm.agcs.data.enablement.DAY0",
    "opcRequestId": "2cec8907-abcd-1234-be17-2dc91122/00ab2d02/2497,
    "tenancyId":
"ocid1.tenancy.oc1..aaaaaaaazp2vvzjsn6newkqrpkwndxpdoixtqfgyhnf4y24h7d5ny27h6f
3q",
    "serviceInstanceId":
"ocid1.agcsgovernanceinstance.oc1.iad.amaaaaaaebkbezqawho7s4aseb4u43vrzy53yiv7
ylgfjqk223wpjc7j4ubq"
  }
}
```

Day 0 Object Export Header Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "object",
  "properties": {
    "eventId": {
      "type": "string"
    },
    "correlationId": {
      "type": "string"
    },
    "eventTime": {
      "type": "string"
    },
    "eventTypeVersion": {
      "type": "string"
    },
    "operation": {
      "type": "string"
    },
    "messageType": {
      "type": "string"
    },
    "status": {
      "type": "string"
    },
    "eventType": {
      "type": "string"
    },
    "opcRequestId": {
      "type": "string"
    },
    "tenancyId": {
      "type": "string"
    },
    "serviceInstanceId": {
      "type": "string"
    }
  },
  "additionalProperties": false
}
```

Sample Output: Day 0 Object Export Header

```
{
  "headers": {
    "eventId": "8787e121-abcd-1234",
    "correlationId": "dc989b5c-abcd-1234",
    "eventTime": "2024-08-27T21:44:15.274034651Z",
    "eventType":
"com.oracle.idm.agcs.data.enablement.policyStatement.created",
    "eventTypeVersion": "1.0",
    "operation": "CREATE",
```

```
    "messageType": "TARGET_ACCESS_POLICY_STATEMENT",
    "status": "IN_PROGRESS",
    "opcRequestId": "2cec8907-abcd-1234-be17-2dc91122/00ab2d02/2497",
    "tenancyId": "ocid1.tenancy.oc1..abcd1234",
    "serviceInstanceId": "ocid1.dev.dev.1234"
  }
}
```

Day N Object Export Header Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "object",
  "properties": {
    "eventId": {
      "type": "string"
    },
    "eventTime": {
      "type": "string"
    },
    "eventTypeVersion": {
      "type": "string"
    },
    "operation": {
      "type": "string"
    },
    "messageType": {
      "type": "string"
    },
    "eventType": {
      "type": "string"
    },
    "opcRequestId": {
      "type": "string"
    },
    "tenancyId": {
      "type": "string"
    },
    "serviceInstanceId": {
      "type": "string"
    }
  },
  "additionalProperties": false
}
```

Sample Output: Day 0 Object Export Header

```
{
  "headers": {
    "eventId": "b8bfe07f-1234",
    "eventTime": "2024-05-18T23:19:56.360412Z",
    "tenancyId": "ocid.tenancy.oc1.1234",
    "serviceInstanceId": "ocid1.instance.oc1.1234",
    "opcRequestId": "request-1234",
    "eventType": "com.oracle.idm.agcs.cloudGroup.created",
```

```
    "eventTypeVersion": "1.0",
    "operation": "CREATE",
    "messageType": "GROUP"
  }
}
```

Header Schema Attribute Definition

Here's the schema for Day 0 and Day N headers available in the output file.

Table 9-1 Header Schema Attribute Definition for Day 0

Attributes	Description
correlationId	Unique identifier to correlate two or more events. For example, if a new resource is created and a new policy grants access to the resource, two events will be published and be identified with this identifier.
eventId	Unique identifier for each event published either to OCI Bucket or OCI Streams. It ensures that each event can be processed and traced distinctly.
eventTime	Timestamp when the event occurred with nanosecond precision. This is required to consume data sequentially and accurately. **Format**: `YYYY-MM-DDTHH:MM:SS.sssssssssZ`
eventTypeversion	Schema version used for sending response for each event. If there are significant changes to schema, then version is updated. For more details, refer Semantic Versioning Guidelines.
messageType	Type of data component being published. Possible values can be • IDENTITY for Identities • GROUP for Identity Collection • RESOURCE for Resource • TARGET_ACCESS_POLICY_STATEMENT for Policies • POLICY_STATEMENT_RESOURCE_MAPPING for Policy to Resource Mapping • RESOURCE_POLICY_STATEMENT_MAPPING for Resource to Policy Mapping
operation	Basic operations associated with the data publishing event. It can be CREATE, UPDATE, DELETE. For some operations, such as policies, if you have to update a policy, events are published with a combination of Create and Delete operations than the update operation.
status	Event Publishing status. Possible values: START, IN PROGRESS, SUCCESS, FAILED. These are sent in the output files. However, on the Oracle Access Governance Console, you can see **Success** or **Failure** status.

Table 9-1 (Cont.) Header Schema Attribute Definition for Day 0

Attributes	Description
eventType	Event value used by the service to track the event operation. For example, if we add a new policy statement in a policy, the value is `com.oracle.idm.agcs.data.enablement.policyStatement.created`
opcRequestId	Unique Oracle-assigned identifier for the request. If you need to contact Oracle about a particular request, please provide the request ID.
tenancyId	Tenancy Oracle Cloud Identifier (OCID) where data is published by .Oracle Access Governance.
serviceInstanceId	Service Instance OCID of your Oracle Access Governance application.

Identity Reference Schema and Sample Output File

Here's Identity schema for creation, modification, and deletion.

Identity Creation Schema

```
{
   "$schema": "https://json-schema.org/draft/2019-09/schema",
   "title": "identities",
   "type": "array",
   "items": {
      "type": "object",
      "properties": {
         "compartmentId": {
            "type": "string"
         },
         "messageType": {
            "type": "string"
         },
         "correlationId": {
            "type": "string"
         },
         "globalIdentity": {
            "type": "object",
            "properties": {
               "identity": {
                  "$ref": "resource:com/oracle/idm/agcs/data/enablement/
schema/identity.json"
               },
               "attributes": {
                  "type": "object",
                  "properties": {
                     "clearance": {
                        "type": "string"
                     },
                     "employeeNumber": {
                        "type": "string"
```

```
                        }
                     }
                  },
                  "id": {
                     "type": "string"
                  },
                  "targetIdentities": {
                     "type": "array",
                     "items": {
                        "properties": {
                           "targetIdentity": {
                              "type": "object",
                              "properties": {
                                 "targetId": {
                                    "type": "string"
                                 },
                                 "identity": {
                                    "$ref": "resource:com/oracle/idm/agcs/data/
enablement/schema/identity.json"
                                 },
                                 "externalId": {
                                    "type": "string"
                                 },
                                 "id": {
                                    "type": "string"
                                 },
                                 "domainId": {
                                    "type": "string"
                                 }
                              }
                           }
                        }
                     }
                  }
               }
            },
            "operation": {
               "type": "string"
            },
            "timestamp": {
               "type": "string"
            }
         },
         "additionalProperties": false,
         "required": [
            "globalIdentity"
         ]
      }
}
```

Identity Modification Schema

```
{
   "$schema": "https://json-schema.org/draft/2019-09/schema",
   "title": "identities",
```

```
   "type": "object",
   "properties": {
      "compartmentId": {
         "type": "string"
      },
      "messageType": {
         "type": "string"
      },
      "correlationId": {
         "type": "string"
      },
      "globalIdentity": {
         "type": "object",
         "properties": {
            "identity": {
               "$ref": "resource:com/oracle/idm/agcs/data/enablement/schema/
identity.json"
            },
            "attributes": {
               "type": "object",
               "properties": {
                  "clearance": {
                     "type": "string"
                  },
                  "employeeNumber": {
                     "type": "string"
                  }
               }
            },
            "id": {
               "type": "string"
            },
            "targetIdentities": {
               "type": "array",
               "items": {
                  "properties": {
                     "targetIdentity": {
                        "type": "object",
                        "properties": {
                           "targetId": {
                              "type": "string"
                           },
                           "identity": {
                              "$ref": "resource:com/oracle/idm/agcs/data/
enablement/schema/identity.json"
                           },
                           "externalId": {
                              "type": "string"
                           },
                           "id": {
                              "type": "string"
                           },
                           "domainId": {
                              "type": "string"
                           }
                        }
```

```
                    }
                 }
              }
           }
        },
        "operation": {
           "type": "string"
        },
        "timestamp": {
           "type": "string"
        }
     },
     "additionalProperties": false,
     "required": [
        "globalIdentity"
     ]
}
```

Identity Deletion Schema

```
{
   "$schema": "https://json-schema.org/draft/2019-09/schema",
   "title": "identities",
   "type": "object",
   "properties": {
      "compartmentId": {
         "type": "string"
      },
      "globalIdentity": {
         "type": "object",
         "properties": {
            "id": {
               "type": "string"
            },
            "targetIdentities": {
               "type": "array",
               "items": {
                  "properties": {
                     "targetIdentity": {
                        "type": "object",
                        "properties": {
                           "id": {
                              "type": "string"
                           }
                        },
                        "additionalProperties": false
                     }
                  }
               }
            }
         },
         "additionalProperties": false
      }
   },
```

```
   "additionalProperties": false,
   "required": [
      "globalIdentity"
   ]
}
```

Sample Output Code Snippet

```
{
   "globalIdentity": {
      "id": "globalId.ICF.EBS_HRMS_STAGE.f014bda5ef4003efa0d8149e59216953",
      "identity": {
         "agStatus": "AG_ACTIVE",
         "agSubType": "WORKFORCE",
         "agRisk": {
            "value": 0,
            "customAttributes": {}
         },
         "agOrganizations": [
            {
               "value": "ba987bed-15ae-47a2-a5b0-265432568ed0",
               "displayName": "PERF_ORGANIZATION-1708541816625"
            },
            {
               "value": "4bf99c0b-ae99-4787-a318-b5eb1e30b89d",
               "displayName": "PERF_ORGANIZATION-1708541866473"
            }
         ],
         "customAttributes": {
            "dateOfBirth": 56160000000,
            "businessGroupId": "7328",
            "supervisorName": "Ivanchuk, Mr. Dmytro",
            "personType": "8351",
            "personId": "28727",
            "grade": "Professional.1",
            "maritalStatus": "S",
            "nationality": "UKR",
            "job": "Buyer",
            "startDate": 828921600000
         },
         "department": "Purchasing",
         "displayName": "Ivan Shevchuk",
         "emails": [
            {
               "value": "Ivan.Shevchuk@example.com"
            }
         ],
         "name": {
            "familyName": "Shevchuk",
            "givenName": "Ivan"
         },
         "organization": {},
         "primaryEmail": "Ivan.Shevchuk@example.com",
         "status": "Active",
         "title": "MR.",
```

```
            "userName": "28727"
         },
         "targetIdentities": [
            {
               "id": "targetId.account.ICF.EBS-
UM.bdf6f156f130553394a859e02f793182",
               "externalId": "1015628",
               "targetId": "a83f87df-75ca-4c4d-966a-2928626e82b8",
               "identity": {
                  "customAttributes": {
                     "operationType": "CREATE_OR_UPDATE",
                     "passwordExpireType": "None",
                     "effectiveStartDate": 1689206400000,
                     "roles": []
                  },
                  "name": {},
                  "primaryEmail": "Ivan.Shevchuk@example.com",
                  "status": "true"
               }
            },
            {
               "id":
"targetId.account.ICF.EBS_HRMS_STAGE.f014bda5ef4003efa0d8149e59216953",
               "externalId": "28727",
               "targetId": "f2a858e5-c449-4a5a-9714-c2e7471b1d2a",
               "identity": {
                  "customAttributes": {
                     "personType": "8351",
                     "title": "MR.",
                     "businessGroupId": "7328",
                     "dateOfBirth": 56160000000,
                     "employeeNumber": "4",
                     "assignments": [
                        {
                           "element": {
                              "organizationId": "7376",
                              "UID": "28936",
                              "jobId": "30930",
                              "gradeId": "18000",
                              "supervisorId": "28725",
                              "effectiveDate": "828921600000"
                           }
                        }
                     ],
                     "hireDate": 828921600000,
                     "gender": "M",
                     "maritalStatus": "S",
                     "operationType": "CREATE_OR_UPDATE",
                     "nationality": "UKR",
                     "lastName": "Shevchuk",
                     "firstName": "Ivan",
                     "addresses": []
                  },
                  "name": {},
                  "primaryEmail": "Ivan.Shevchuk@example.com",
                  "status": "true"
```

```
                }
            }
        ]
    }
}
```

Identity Schema Attribute Definition

Here's the attribute definition for an identity export file.

Table 9-2 Identity Schema Attribute Definition for Day 0

Attributes	Description
globalIdentity	Composite identity profile object used by Oracle Access Governance as a source of truth to perform various governance and provisioning operations. It contains access profile details, including core and custom attributes. For more information, refer to Identities Access Details Reference,
globalIdentity → id	Unique identifier for the resource within Oracle Access Governance. This also includes the orchestrated system information from where the resource value is ingested.
targetIdentities	Orchestrated identity object integrated with Oracle Access Governance and matched with the composite identity profile.
targetIdentities → id	Unique identifier for the resource within Oracle Access Governance. In this case, it depicts orchestrated system integrated with Oracle Access Governance. This includes the orchestrated system name identifier.
targetId	Unique identifier for the orchestrated system integrated with Oracle Access Governance.

Identity Collection Reference Schema and Sample Output File

Here's Identity Collection schema for creation, modification, and deletion.

Identity Collection Creation Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string"
      },
      "domainId": {
        "type": "string"
      },
      "compartmentId": {
        "type": "string"
```

```
},
"externalId": {
  "type": "string"
},
"targetId": {
  "type": "string"
},
"name": {
  "type": "string"
},
"description": {
  "type": "string"
},
"add": {
  "type": "object",
  "properties": {
    "identities": {
      "type": "array",
      "items": {
        "properties": {
          "targetIdentity": {
            "type": "object",
            "properties": {
              "targetId": {
                "type": "string"
              },
              "externalId": {
                "type": "string"
              },
              "targetIdentityId": {
                "type": "string"
              }
            }
          }
        }
      }
    }
  }
},
"remove": {
  "type": "object",
  "properties": {
    "identities": {
      "type": "array",
      "items": {
        "properties": {
          "targetIdentity": {
            "type": "object",
            "properties": {
              "targetId": {
                "type": "string"
              },
              "externalId": {
                "type": "string"
              },
              "targetIdentityId": {
```

```
                    "type": "string"
                  }
                }
              }
            }
          }
        }
      }
    }
  },
  "additionalProperties": false,
  "required": [
    "id"
  ]
}
```

Identity Collection Modification Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string"
      },
      "domainId": {
        "type": "string"
      },
      "compartmentId": {
        "type": "string"
      },
      "externalId": {
        "type": "string"
      },
      "targetId": {
        "type": "string"
      },
      "name": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "add": {
        "type": "object",
        "properties": {
          "identities": {
            "type": "array",
            "items": {
              "properties": {
                "targetIdentity": {
                  "type": "object",
```

```
                    "properties": {
                      "targetId": {
                        "type": "string"
                      },
                      "externalId": {
                        "type": "string"
                      },
                      "targetIdentityId": {
                        "type": "string"
                      }
                    }
                  }
                }
              }
            }
          }
        },
        "remove": {
          "type": "object",
          "properties": {
            "identities": {
              "type": "array",
              "items": {
                "properties": {
                  "targetIdentity": {
                    "type": "object",
                    "properties": {
                      "targetId": {
                        "type": "string"
                      },
                      "externalId": {
                        "type": "string"
                      },
                      "targetIdentityId": {
                        "type": "string"
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    },
    "additionalProperties": false,
    "required": [
      "id"
    ]
}
```

Identity Collection Deletion Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
```

```
  "title": "identities",
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    }
  },
  "additionalProperties": false,
  "required": [
    "id"
  ]
}
```

Sample Output Code Snippet

```
{
  "id": "group.OCI.accessgovtest.632e77bf5a9595695e1d8ec629c0a32a",
  "domainId": "resource.OCI.accessgovtest.499e3f20709d30c915ff95f686b9b4e0",
  "externalId":
"ocid1.group.oc1..aaaaaaaa45mrsaj4pz22vjs5avgn3uvdqszgihpxic45aenjjugyevrizmtq
",
  "targetId": "9dff2808-ea31-41a3-81ed-5800190acf38",
  "compartmentId":
"resource.OCI.accessgovtest.194ec6f9cb821ab9aaf075f0e7f42bc2",
  "name": "group-events",
  "remove": {
    "identities": []
  },
  "add": {
    "identities": [
      {
        "externalId": "ff016cee1a8b4739bde4eb080c5b0b19",
        "targetIdentityId":
"targetId.account.OCI.accessgovtest.1810d44f39cf1bb7913e0ac3941fcaab",
        "id": "globalId.june-stage-qa1-
agent.29025.kbezqawho7s4aseb4u43vrzy53yiv7ylgfjqk223wpjc7j4ubq"
      },
      {
        "externalId": "ff09a2c5bee34be0ad88564381f93fbd",
        "targetIdentityId":
"targetId.account.OCI.accessgovtest.2938067570ac7dea662f5978e49fa4fd",
        "id": "globalId.ICF.EBS_HRMS_STAGE.b9c25ec7b8b5cbf9aeaa000f204a36d3"
      },
      {
        "externalId": "fdbeefb62f0d4923b2bcd1ae1e657924",
        "targetIdentityId":
"targetId.account.OCI.accessgovtest.c562efca9023e59e798ef1d544bf0ce1",
        "id": "globalId.OCI.accessgovtest.c562efca9023e59e798ef1d544bf0ce1"
      },
      {
        "externalId": "fec6a739324843ecbc7d6add45180b58",
        "targetIdentityId":
"targetId.account.OCI.accessgovtest.d7ba12539289eaff44e2ea3b22297dc3",
        "id": "globalId.OCI.accessgovtest.d7ba12539289eaff44e2ea3b22297dc3"
      },
```

```
      {
        "externalId": "fdabd17eb42f47369b81aa66884162ff",
        "targetIdentityId":
"targetId.account.OCI.accessgovtest.025820803c0bad2da4da49f1df78e258",
        "id": "globalId.OCI.accessgovtest.025820803c0bad2da4da49f1df78e258"
      },
      {
        "externalId": "fe9ac7d01ae84cbb829bf08ddff1a869",
        "targetIdentityId":
"targetId.account.OCI.accessgovtest.a6ba9c0697027906f1396935714da8c5",
        "id": "globalId.OCI.accessgovtest.a6ba9c0697027906f1396935714da8c5"
      }
    ]
  }
}
```

Identity Collection Schema Attribute Definition

Here's the attribute definition for an identity collection export file. You'll only be able to publish OCI group details.

Table 9-3 Identity Collection Schema Attribute Definition

Attributes	Description
id	Unique identifier for the resource within Oracle Access Governance. This also includes the orchestrated system information from where the resource value is ingested.
domainId	Unique domain identifier (OCID) associated with the identity collection (IAM group) ingested into Oracle Access Governance. This is applicable only for OCI orchestrated system and contains OCI IAM groups.
externalId	Refers to OCID of the object on the OCI console. For an OCI group, the external id may look like `ocid1.group.oc1.ab1234a`
targetId	Unique identifier for the orchestrated system integrated with Oracle Access Governance.
compartmentId	Unique compartment identifier (OCID) associated with the identity collection. This is applicable only for OCI orchestrated system and contains OCI IAM groups.
name	Identity collection name.
description	Identity collection description.
add	Array of identities included in the identity collection.
remove	Array of identities excluded from this identity collection. Update and Create operations share the same schema. However, when a new identity collection is created, you'll not receive any identities in this attribute.

Policies Reference Schema and Sample Output File

Here's Policies schema for creation and deletion.

Policies Creation Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {
        "type": "string"
      },
      "compartmentId": {
        "type": "string"
      },
      "externalId": {
        "type": "string"
      },
      "targetId": {
        "type": "string"
      },
      "policyStatementId": {
        "type": "string"
      },
      "name": {
        "type": "string"
      },
      "description": {
        "type": "string"
      },
      "statement": {
        "type": "string"
      },
      "subjects": {
        "type": "array",
        "items": {
          "properties": {
            "id": {
              "type": "string"
            },
            "name": {
              "type": "string"
            },
            "type": {
              "type": "string"
            }
          }
        }
      },
      "verb": {
        "type": "string"
```

```
      },
      "resourceType": {
        "type": "string"
      },
      "location": {
        "type": "object",
        "properties": {
          "compartment": {
            "type": "string"
          }
        }
      },
      "tags": {
        "type": "object"
      }
    }
  },
  "additionalProperties": false,
  "required": [
    "id"
  ]
}
```

Policies Modification Schema

Policy Modifications are handled using a combination of create and delete operations. To update a policy, existing policy is first deleted before replacing it with a policy with new parameters.

Policies Deletion Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "additionalProperties": false,
    "required": [
      "id"
    ]
  }
}
```

Sample Output Code Snippet

```
{
  "headers": {
    "eventId": "8788h161-acde-43a4-87e4-b6f01ca3aaf0",
    "correlationId": "dc98e55c-f574-422f-a9ce-326fce9d9edc",
    "eventTime": "2024-08-27T21:44:15.274034651Z",
    "eventType":
"com.oracle.idm.agcs.data.enablement.policyStatement.created",
    "eventTypeVersion": "1.0",
```

```
    "operation": "CREATE",
    "messageType": "TARGET_ACCESS_POLICY_STATEMENT",
    "status": "IN_PROGRESS",
    "opcRequestId": <opc-request-id>,
    "tenancyId": <tenancy-id>,
    "serviceInstanceId": <service-instance-ocid>
  },
  "data":
"[{\"id\":\"tapolicy.OCI.agcusttokyo.aed5bbcee30da0a828e76f01deef7090\",\"exte
rnalId\":\"ocid1.policy.oc1..aaaaaaaabgvxsco3avg772ehflmwvljaur75zvzdwh6y7wqhz
hvcvp2mxtha\",\"targetId\":\"fdb6c5f7-6e3a-4f36-9dd0-
a17993be389f\",\"policyStatementId\":\"tapolicystmt.OCI.agcusttokyo.a72df097dc
e1deecf8606c59b6dec588\",\"name\":\"DummyPolicy20Nov\",\"description\":\"Dummy
Policy20Nov\",\"statement\":\"Allow group \\u0027TestAlpha\\u0027/\
\u0027ComputeGroup\\u0027 to manage instance-family in tenancy\",\"subjects\":
[{\"id\":\"TestAlpha/ComputeGroup\"}],\"verb\":\"MANAGE\",\"resourceTypes\":
[\"VolumeAttachment\",\"InstanceConsoleConnection\",\"Instance\",\"AppCatalogL
isting\",\"ComputeCapacityReservation\",\"DedicatedVmHost\",\"AutoScalingConfi
guration\",\"InstanceAgentCommand\",\"ConsoleHistory\"],\"location\":
{\"compartment\":\"agcusttokyo\"}},
{\"id\":\"tapolicy.OCI.agcusttokyo.aed5bbcee30da0a828e76f01deef7090\",\"extern
alId\":\"ocid1.policy.oc1..aaaaaaaabgvxsco3avg772ehflmwvljaur75zvzdwh6y7wqhzhv
cvp2mxtha\",\"targetId\":\"fdb6c5f7-6e3a-4f36-9dd0-
a17993be389f\",\"policyStatementId\":\"tapolicystmt.OCI.agcusttokyo.08940cfb6d
b80a7d9b4027e3c9994d51\",\"name\":\"DummyPolicy20Nov\",\"description\":\"Dummy
Policy20Nov\",\"statement\":\"Allow group \\u0027TestAlpha\\u0027/\
\u002726DecCloudCompute\\u0027 to read app-catalog-listing in
tenancy\",\"subjects\":[{\"id\":\"TestAlpha/
26DecCloudCompute\"}],\"verb\":\"READ\",\"resourceTypes\":
[\"AppCatalogListing\"],\"location\":{\"compartment\":\"agcusttokyo\"}}]"
}
```


Policies Schema Attribute Definition

Here's the attribute definition for policy export file.

Table 9-4 Policy Schema Attribute Definition

Attributes	Description
id	Unique identifier for the policy assigned within Oracle Access Governance.
compartmentId	Unique compartment identifier (OCID) associated with the policy. This is applicable only for OCI policies.
externalId	Unique policy identifier in OCI, called OCID. For policy, the external id may look like `ocid1.policy.oc1.aa1234`
policyStatementId	Unique identifier for each policy statement associated with the policy.
name	Policy name.
description	Policy description
statement	Policy rules governing control of resources. Each policy consists of one or more policy statements
subjects	Array of principals to which the access is granted by this policy, for example, IAM group-name.
verb	Access grant type assigned to a resource by using verbs in the policy. Possible verbs may be `inspect`, `read`, `use`, `inspect`. `manage`.

Table 9-4 (Cont.) Policy Schema Attribute Definition

Attributes	Description
resourceType	Array of resource types associated with a policy. It can be family resource-type or individual resource-type. For example, `instance`, `volumes`, `volume-family`, and so on. For more information, see Resource Types in OCI.
location	Scope of access granted through this policy, such as specific compartment or entire tenancy.

Resource Reference Schema and Sample

Here's resource schema for creation, modification, and deletion.

Resource Creation Schema

```
{
   "$schema": "https://json-schema.org/draft/2019-09/schema",
   "title": "resources",
   "type": "array",
   "items": {
      "type": "object",
      "properties": {
         "id": {
            "type": "string"
         },
         "externalId": {
            "type": "string"
         },
         "targetId": {
            "type": "string"
         },
         "tenancyId": {
            "type": "string"
         },
         "resourceName": {
            "type": "string"
         },
         "resourceType": {
            "type": "string"
         },
         "description": {
            "type": "string"
         }
      },
      "additionalProperties": false
   }
}
```

Resource Modification Schema

```
{
   "$schema": "https://json-schema.org/draft/2019-09/schema",
   "title": "resources",
```

```
   "type": "array",
   "items": {
      "type": "object",
      "properties": {
         "id": {
            "type": "string"
         },
         "externalId": {
            "type": "string"
         },
         "targetId": {
            "type": "string"
         },
         "tenancyId": {
            "type": "string"
         },
         "resourceName": {
            "type": "string"
         },
         "resourceType": {
            "type": "string"
         },
         "description": {
            "type": "string"
         }
      },
      "additionalProperties": false
   }
}
```

Resource Deletion Schema

```
{
   "$schema": "https://json-schema.org/draft/2019-09/schema",
   "type": "object",
   "properties": {
      "id": {
         "type": "string"
      },
      "additionalProperties": false,
      "required": [
         "id"
      ]
   }
}
```

Sample Output Code Snippet

```
{
   "id": "resource.ICF.ADUPDATE.4bbac5904b6302dc82871da2c9756fea",
   "externalId": "48713388",
   "targetId": "ade93352-e7d5-46e6-847a-c765be1f0aad",
   "tenancyId":
"ocid1.tenancy.oc1..aaaaaaaahvjxelu7yccuhj3wrq5uqiybu7f5tfxvwteiwaupnlkj4woz6y
bq",
```

```
   "resourceName": "ADUPDATE",
   "resourceType": "AD",
   "description": ""
}
```

Resources Schema Attribute Definition

Here's the attribute definition for an resource export file.

Table 9-5 Resource Schema Attribute Definition

Attributes	Description
id	Unique identifier assigned within Oracle Access Governance for resource tracing. It also contains orchestrated system identifier from which the resource is ingested into Oracle Access Governance.
externalId	Unique resource identifier in OCI.
targetId	Unique identifier for the orchestrated system integrated with Oracle Access Governance.
tenancyId	Unique tenancy identifier (OCID) in which the resource is located. This is applicable only for OCI orchestrated system and contains OCI resources.
resourceName	Resource name.
resourceType	Resource Type
description	Resource description

Resource to Policy Statement

Here's a schema for list of policies associated with a resource.

Policy Statement to Resource Creation Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "title": "accessPolicyStatementResourceMapping",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "compartmentId": {
        "type": "string"
      },
      "id": {
        "type": "string"
      },
      "externalId": {
        "type": "string"
      },
      "targetId": {
        "type": "string"
      },
      "policies": {
        "type": "array",
        "items": {
          "type": "object",
```

```
          "properties": {
            "id": {
              "type": "string"
            },
            "externalId": {
              "type": "string"
            },
            "policyStatementId": {
              "type": "string"
            }
          }
        }
      }
    }
  }
}
```

Sample Output Code Snippet

```
{
  "compartmentId":
"ocid1.tenancy.oc1..ppppppp2h5y42lkaalhtrwmqyinmwipjyxc3xmod4h7m3d2bmdjg6qwert
y",
  "id": "resource.OCI.agcusttokyo.1b65a16c154269702eea873f34cef690",
  "externalId": "ocid1.database.oc1.ap-
tokyo-1.anxhiljrzqwertya7o46ijh4nv3rjzpnqjwqidqh37rcptyngy5g46ebnlea",
  "targetId": "e88d075e-d2a6-4f1d-8c1b-f472917b8770",
  "policies": [
    {
      "id": "tapolicy.OCI.agcusttokyo.70ffb4c4f706aa55a5a35cb7902fe47a",
      "externalId":
"ocid1.policy.oc1..aaaaaaaaxkyqwertyenond5hoclrmmvhlxw3tjukgqbbstfmepigetr6ulq
q",
      "policyStatementId":
"tapolicystmt.OCI.agcusttokyo.99cd276ef37300a357c0a1488dae2567"
    }
  ]
}
```

Resources to Policy Schema Attribute Definition

Here's the attribute definition for an identity export file.

Table 9-6 Resource to Policy Schema Attribute Definition

Attributes	Description
compartmentId	Unique compartment identifier (OCID) associated with the resource. This is applicable only for OCI resources.
id	Unique identifier for the resource assigned within Oracle Access Governance.
externalId	Unique resource identifier in OCI, called resource OCID.
targetId	Unique identifier to identify orchestrated system associated with the resource.

Table 9-6 (Cont.) Resource to Policy Schema Attribute Definition

Attributes	Description
policies	Array of policies attached to a resource. Each policy contains details like policy id, policy statement id, and external id to identify policies

Policy Statement to Resource

Here's a schema for a policy statement associated with a list of resources.

Policy Statement to Resource Creation Schema

```
{
  "$schema": "https://json-schema.org/draft/2019-09/schema",
  "title": "accessPolicyStatementResourceMapping",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "compartmentId": {
        "type": "string"
      },
      "id": {
        "type": "string"
      },
      "externalId": {
        "type": "string"
      },
      "policyStatementId": {
        "type": "string"
      },
      "targetId": {
        "type": "string"
      },
      "resources": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "id": {
              "type": "string"
            },
            "externalId": {
              "type": "string"
            }
          }
        }
      }
    }
  }
}
```

Sample Output Code Snippet


```
{
  "compartmentId":
"ocid1.tenancy.oc1..aaaaaaaazp2vvzjsn6newkqrpkwndxpdoixtqfgyhnf4y24h7d5ny27h6f
3q",
  "id": "tapolicy.OCI.ag-test.58a173b97f94c6ff0046448470573609",
  "externalId":
"ocid1.policy.oc1..aaaaaaaa3axgfgqgc3f4xcbno5p7driojc2hnaxr3jw3pu5xs7lhkoopo55
a",
  "policyStatementId": "tapolicystmt.OCI.ag-
test.2dc34ea12d5f0cbd7e9429029b257c99",
  "targetId": "2879c2b3-721c-4ede-afc5-5aa6c3b54e99",
  "resources": [
    {
      "id": "resource.OCI.ag-test.189fa43b2323037d1da11e6f0e488296",
      "externalId": "ocid1.instance.oc1.me-
abudhabi-1.anqxkljrebkbezqcpoofyvho44qcyb76uu75aonuhexk42ibcv4ohpfat5iq"
    },
    {
      "id": "resource.OCI.ag-test.s563541re3fca34e7105fe5a7c516025",
      "externalId":
"ocid1.instance.oc1.iad.anuwcljsebkbezqcowg5so6mnuewanlrohoovt7ce54ujhuqxi3h75
hm6mrq"
    },
    {
      "id": "resource.OCI.ag-test.d3e8c10a5659d18dda403bf00e1e2aea",
      "externalId":
"ocid1.instance.oc1.iad.anuwcljtebkbezqcxvwlq62t4ldkyb5kkttgvyobqvuh3tltz7qjtx
chldja"
    },
    {
      "id": "resource.OCI.ag-test.c189f74e4c77ca6416f0d92bb2db9f2a",
      "externalId": "ocid1.instance.oc1.us-
sanjose-1.abcdejrebkbezqctftv5vbfucdb3amdgslvbptnjpdqyvporupbhr3sluqq"
    },
    {
      "id": "resource.OCI.ag-test.a1f0662fd624e07c6b41e14fdf918591",
      "externalId":
"ocid1.instance.oc1.iad.abcdejtebkbezqctyoxut22u26pvej5wglyodyjen6fb7qqid3ovy4
sfbxa"
    },
    {
      "id": "resource.OCI.ag-test.8d177901e639e144b6289ed1d81fe255",
      "externalId":
"ocid1.instance.oc1.iad.qwertyjtebkbezqcoxwhqw4wfpmlozp5ghrhr74222jcivbdaannyo
6a26xq"
    },
    {
      "id": "resource.OCI.ag-test.81bdacf188ed585e3aca85a131e23df5",
      "externalId":
"ocid1.instance.oc1.iad.qwertyjtebkbezqcdtejri54el6u7dw4bvjmz5t2rzxar76oz5szig
f2o4aq"
    },
    {
```

```
      "id": "resource.OCI.ag-test.3883fc488ff0531274fcee9d87f00fd2",
      "externalId":
"ocid1.instance.oc1.iad.qwertyjtebkbezqczdrg677joyhxf2kdm2jzlzmetzzylox4lptxd2
svbnva"
    },
    {
      "id": "resource.OCI.ag-test.3480fad0d9bf8a3fe13c54028f13f66c",
      "externalId": "ocid1.instance.oc1.us-
sanjose-1.qwertyjrebkbezqcyrzc7xu3flvidhwoyssaoiyyewi3bidnnnuucpfjmwwq"
    },
    {
      "id": "resource.OCI.ag-test.3480fad0d9bf8a3fe13c54028f13f66c",
      "externalId": "ocid1.instance.oc1.us-
sanjose-1.qwertyjrebkbezqcyrzc7xu3flvidhwoyssaoiyyewi3bidnnnuucpfjmwwq"
    },
    {
      "id": "resource.OCI.ag-test.2a14a90e547488318d2bca0b30a247f1",
      "externalId":
"ocid1.instance.oc1.ad.qwertyjtebkbezqcwyajizhztiujrmmshn3cuqiou4vtodhv4femidl
pj7ha"
    }
  ]
}
```


Policy to Resources Schema Attribute Definition

Here are the definitions of the attribute included in the policy access to resource export file.

Attributes	Description
compartmentId	Unique compartment identifier (OCID) associated with a policy. This is applicable only for OCI policies.
id	Unique identifier for the policy assigned within Oracle Access Governance.
externalId	Unique policy identifier in OCI, called OCID.
targetId	Unique identifier to identify orchestrated system associated with the policy.
resources	Array of resources attached to a resource. Each policy contains details like policy id, policy statement id, and external id to identify policies

10
Reference

A

Access Bundle
A collection of permissions associated with an application or service. Access Bundles are used to assign permissions to identities through a policy or by request.

Access Control
A mechanism to govern (approve or revoke) access privileges, such as permissions, accounts and role membership assigned to users, or access controls that grant access.

Access Profile (My Access)
All access information associated with an identity, including their accounts, set of permissions, and role memberships, representing the logical level of access over applications, service and/or resource.

Access Review campaigns
The mechanism to govern (approve or revoke) the access privileges such as permissions, accounts and role membership assigned to users, and also govern the access privileges associated with OCI IAM Policies.

Account
A representation created in Managed Systems that may include access permissions for resources in the Managed system. Account attributes can be account name, description, status and so on. Accounts could be associated with an Identity either through provisioning process or through discovery process. Any unassociated account would be marked as an **Unmatched** account that could be a rogue account or a valid orphaned account.

Attributes
The data elements that store information related to an object such as Identity, Role, and Permission. For example, the attributes for an identity are first name, last name, department, manager, cost center, position, and email address.

Authoritative Source
A repository or system that contains identity information and is considered to be the primary or most reliable source for this information. For example, HR system (such as Fusion Apps HCM, Oracle EBS, PeopleSoft) for user attributes such as employee's first name, last name, department, manager, cost center, position, and email address.

C

Campaign Selection Criteria
The set of rules that defines the scope of an access review campaign.

Campaign Owner
A user in Oracle Access Governance that has special permissions to manage the access review campaigns they own. Campaign owner is defined while creating an access review campaign.

Correlation
The process to determine whether an ingested account or an identity belongs to the existing identity to build a composite identity profile.

Consumer User
An identity who is either an individual, such as customers, alumni, and outsourced partners, or as a service identity, such as devices which are configured not to access the Access Governance service during the billing period, regardless of whether the individual or service is actively accessing the hosted service at any given time.

D

De-provisioniong
The process of removing user access to an application, service, software system, or hardware. This process happens automatically when access permissions are revoked during an access review campaign, or when a role membership or permission is revoked through an access policy in Oracle Access Governance.

Delegations
Temporary transfer of responsibility for completing a task, such as performing access reviews, or approving request) to another Oracle Access Governance active user. The original assignee still retains the ownership.

Data Feed
Data Feed is an Oracle Access Governance service used to send data events to an external system. A data event can be an update related to Oracle Access Governance data components, such as creation of new account, modification in resources, or alteration to policy. Data Feed publishes real-time updates as a continuous stream in a sequential order.

E

Event-based access reviews
Continuous access reviews that are triggered when user attributes such as organization, department, manager, location, are updated.

Event Data Publishing
Process to export and continually publish data events in real-time to external systems.

G

Grant Type
A method used to provision access to identities for a specific resource. These resources can be provisioned directly, through a policy, or can be requested by an identity.

I

Identity
A unique representation of a user or machine in Oracle Access Governance, with attributes like

first name, last name, username, email and other attributes sourced from one or multiple Authoritative systems.

Identity Collection
A set of identities, created to assign access privileges over applications and resources to its members.

Identity Orchestration
Oracle Access Governance brings together diverse Authoritative Sources and Managed Systems by supporting low-code integrations. It facilitates data transformations and correlation rules which ensures data coherence, extracts the required identity data from various systems into Oracle Access Governance, enables businesses to perform robust access control, intelligent access reviews, and perform fulfillment through account provisioning.

Identity Hub
The Identity Orchestration engine of Oracle Access Governance that fetches or reconciles identity and access data from identity orchestration, and provisions identity and access data from Oracle Access Governance to the identity orchestration.

Inbound Data Transformations
Inbound data transformations allow you to modify identity or account data values during the data ingestion process.

Insights & Recommendations
A set of prescriptive analytics and identity intelligence from identity and access data, enabling access reviewers and approvers to take quick and correct actions efficiently.

J

Joiner-Mover-Leaver
Refers to the different types of provisioning supported by an Identity Governance and Administration service.

- *Joiner* refers to action taken by the system when an identity joins the company, such as assigning some birth-right access privileges.
- *Mover* refers to the acton taken by the system when an identity moves within the same organization, for example, changes in access privileges when user changes location or job.
- *Leaver* refers to the actions taken by the system when an identity leaves the company, such as revoking access over all corporate applications and systems.

M

Managed System
Applications and services containing accounts and respective access privileges but do not serve as a trusted source of identities in your enterprise information. By establishing an orchestrated system, Oracle Access Governance manages user accounts and access permissions for these applications leveraging the defined access controls.

Matching Rules
Also called Correlation Rules. See Correlation .

N

Notification

Automated email alerts to keep you informed of significant events occurring within the Oracle Access Governance service instance. These resources can be related to account operations, approval operations, review tasks, or error alerts.

O

Overview
The first page users see when they log in into the Oracle Access Governance service. This page shows the widgets available to users to track their access privileges, access review and approval tasks.

Orchestrated System
Oracle Access Governance can be integrated with various applications and systems. These systems can be authoritative sources of identity data (for example, HR systems, Active Directory) or managed systems in which access privileges are granted (for example, applications, databases).

Orphan Account
An identity account not associated with any active identity.

Organization
Logical and hierarchical grouping of identities, such as belonging to same business unit, to control access management and access reviews operations within an enterprise.

P

Permission
A specialized type of assignment that defines access rights and the set of actions an identity can perform over specified resources and applications, for example, access to some sections in Oracle Access Governance console.

Policy
Groups together multiple identity collections, roles, and access bundles in associations

Provisioning
The process of adding user's access to an application, service and/ or software system, or hardware. Provisioning occurs automatically when certain access permissions are approved, or when some role membership or permission is assigned to a user through an access policy in Oracle Access Governance.

R

Resource & Application
The external system, cloud service, database, directory server, or other source of identity data to be managed and audited by an identity management system.

Role
A collection of permissions and access bundles associated with one or multiple applications or services. Roles are used to assign permissions to identities through policies, or by request.

S

Service Account
The administrative account or any account on any system that manages that system. It can be assigned to an Identity or Identity Collection.

Service Instance
With respect to Oracle Access Governance, Service Instance refers to cloud application instance running on Oracle Cloud Infrastructure (OCI). Each instance is uniquely identified by an Oracle Cloud Identifier (OCID), along with compartment, region, license type, allowing you to manage these across cloud environment. Oracle Access Governance service instance have a format `AG-<servicename>`, having an `AG` prefix.

W

Workflow
A business process that orchestrates end-to-end activities, involving sequential and parallel steps, through which an access request or review passes, from initiation to completion.

Workforce User
An identity which is either an individual, such as an employee or contractor, or a service identity, such as bots,applications, or services, which is configured to access the Oracle Access Governance service during the billing period, regardless of whether the individual or service is actively accessing the hosted service at any given time.

Supported Languages in Oracle Access Governance

You can view Oracle Access Governance Console in your preferred language based on your browser's locale settings. It currently supports 34 different languages. Update your browser locale settings to switch the view to your preferred language from your browser language settings.

Language and Locale

Oracle Access Governance supports the following languages:

Table 10-1 Supported UI Languages

Language	Locale
Arabic	`ar`
Arabic - Bidirectional	`ab-XB`
Chinese - Simplified	`zh-Hans (or zh-CN)`
Chinese - Traditional	`zh-Hant (or zh-TW)`
Croatian	`hr`
Czech	`cs`
Danish	`da`
Dutch	`nl`
English	`en`
English-Psuedo (Latin)	`en-XA`
English-Psuedo (Mixed)	`en-XC`
Finnish	`fi`
French-Canada	`fr-CA`

Table 10-1 (Cont.) Supported UI Languages

Language	Locale
French - France	fr
German	de
Greek	el
Hebrew	he
Hungarian	hu
Italian	it
Japanese	ja
Korean	ko
Norwegian	no
Polish	pl
Portuguese - Brazil	pt-BR
Portuguese - Portugal	pt
Romanian	ro
Russian	ru
Serbian - Cyrillic	sr
Slovak	sk
Slovenian	sl
Spanish - Worldwide	es
Swedish	sv
Thai	th
Turkish	tr

Switching to the Preferred Language in your Browser

You will view Oracle Access Governance Console in the language based on the browser's locale settings. It automatically detects the preferred language set in your browser, and adjusts the content accordingly. If the browser's locale is set to French, you can view Oracle Access Governance in French. You can change your preferred language by changing the default locale in your browser. For more information, refer to the browser's documentation to change your language settings.

Here are a few reference links for changing language settings for the popular browsers:

- **Google Chrome**: Change Chrome Language
- **Mozilla Firefox**: Change Firefox Language
- **Microsoft Edge**: Change Edge Language
- **Safari** (on MacOS): Change Safari Language

> **Note:**
>
> These are external links, and their content or URL may change. If you are unable to visit any link, we recommend you to search online for the latest instructions.