

Oracle CMU Implementation Checklist

Avoid Common Active Directory Integration and Role Design Mistakes.

Your step-by-step guide to implementing Oracle Centrally Managed Users (CMU) correctly.

Stop managing Oracle users one database at a time.

If you manage Oracle users across multiple databases, you're spending more time on password resets and access requests than you should. Oracle Centrally Managed Users (CMU) centralizes authentication in Active Directory. This checklist makes sure you configure it right.

Work through each phase in order. Skip nothing.

Phase 1: Architecture and Prerequisites

Understand what you are getting into

Before touching any configuration: plan your schema strategy and Active Directory (AD) group design first. Retrofitting these decisions later costs far more time than planning them now.

	Task	What to verify
☐	Oracle version	Requires Oracle Database 18c or later. Run: <code>SQL> SELECT version_full FROM v\$instance</code>
☐	AD connectivity	Database host must reach AD on LDAPS (TLS) port 636. Test with <code>ldapsearch</code> or <code>telnet</code> before anything else.
☐	AD service account	Create a dedicated read-only service account in AD for your Oracle database. Document it. Do not use a personal account.
☐	Authentication method	Decide on authentication method, password or Kerberos. Password authentication is the most convenient and compatible method.
☐	Schema strategy	Shared schema or exclusive schema? Shared = less DDL, scales better. Exclusive = full schema isolation, more maintenance.
☐	AD group design	Map AD security groups to Oracle global roles. Design this mapping on paper before writing a single line of SQL.

	Task	What to verify
☐	DBA and AD team aligned	Both teams must be active in this project. CMU spans two domains. Misalignment here causes the most delays.

PRO TIP	Shared schemas are the right default for most multi-database environments. One schema per business function, not one schema per user. You can always add exclusive schemas later for exceptions.
----------------	--

PRO TIP	Password authentication is compatible with every application that may connect to your Oracle database whereas Kerberos authentication may not be. Also, Kerebos authentication may add extra steps for the user in the login process.
----------------	---

Phase 2: Active Directory Configuration for Password Authentication

Extend Active Directory for Oracle Database Password Authentication

Work through these steps in order. Each one is a dependency for the next.

	Task	What to verify
☐	Get Latest Extension	Download the latest extension utility from My Oracle Support (MOS)
☐	AD Extension installation	Run the extension utility on each domain controller. Note, the domain controller needs to be rebooted after the install.
☐	AD User Configuration	Add Oracle database users in AD to the ORA_VFR_12C group, the group is created by the install tool. The user will need to change their password after this is done.
☐	Create Service Account	Create a read only service account in AD that will be used by the database to lookup users and groups.

PRO TIP	Activate Directory (AD) extensions are a normally supported process by Microsoft, and the Oracle password extension has been available for over 20 years and is widely used by Oracle customers.
----------------	--

Phase 3: Oracle and Active Directory Configuration

Connect Oracle to Active Directory

Work through these steps in order. Each one is a dependency for the next. A misconfigured dsi.ora or wallet will produce cryptic errors that are hard to trace.

	Task	What to verify
☐	Oracle Wallet	Create with orapki and mkstore. Stores the AD bind credentials as well as the TLS certificate. Place it in a dedicated directory where the Oracle OS user has read access.
☐	Create Directory Object	Create a directory object in the database where the wallet is located.
☐	dsi.ora	Place this file in the same directory as the wallet. It lists the AD server names and search path in AD.
☐	Configure Database	Set database parameter LDAP_DIRECTORY_ACCESS and property CMU_WALLET
☐	Service account verified	Run ldapsearch -H ldaps://adhost:636 -D bindDN -W -b baseDN to confirm Oracle can query AD before proceeding.
☐	DNS resolution	Database host must resolve the AD domain controller by FQDN. Confirm with nslookup or dig. LDAP failures here are silent.

Phase 4: Database Schema and Roles Configuration

Configure Oracle to authenticate via AD

These are schema and roles changes inside Oracle. Test each one on a non-production database.
Note: app_dev and dev_role are sample names.

	Task	What to verify
☐	Global shared schema	CREATE USER <i>app_dev</i> IDENTIFIED GLOBALLY AS "CN=ora_app_dev_group,CN=Groups,DC=mycompany,=DC=com"; This points to a group in AD, all members now have access to the database.
☐	Global roles	CREATE ROLE <i>dev_role</i> IDENTIFIED GLOBALLY AS 'CN=ora_app_dev_group,CN=Groups,DC=mycompany,=DC=com'; One role per AD group. Keep role names consistent with your AD group naming convention. Grants to this role are given to all members, and rights from multiple groups are additive.
☐	Privilege grants	Grant object privileges to the schema, either shared or exclusive. Or rely on global roles for all grants.
☐	Enable Users in AD	Put users into groups in AD mapping to shared schema and global roles in Oracle as appropriate.

	Task	What to verify
☐	Test authentication	Use the users AD name to login such as: first.last@tnsalias. Verify with: <pre>SELECT sys_context('USERENV', 'AUTHENTICATED_IDENTITY') FROM DUAL;</pre>
☐	Role verification	After login, confirm the AD group-to-Oracle-role mapping resolved correctly, run: <pre>SELECT * FROM SESSION_ROLES;</pre>

PRO TIP

If the test login fails with ORA-01017 (Invalid Username or Password), the wallet, dsi.ora, database, or AD groups are misconfigured.

Phase 5: Security Hardening

Lock it down before go-live

CMU moves your authentication perimeter into Active Directory. Your AD security posture now directly affects Oracle access. These are not optional steps.

	Task	What to verify
☐	LDAPS enforced	Use LDAPS (port 636) with a valid TLS certificate in production. Plain LDAP is a credential exposure risk.
☐	Bind account hardened	Bind account must have read-only access to user and group objects only. No password reset, no group membership changes.
☐	Oracle Wallet permissions	Wallet directory: chmod 700. Wallet files: chmod 600. Owner must be the Oracle OS user. No world-readable wallet files.
☐	AD password policy	Confirm AD password expiration is communicated to Oracle users. Expired AD credentials lock Oracle access with no Oracle-side warning.
☐	Unified Auditing	Enable audit policy for logon events and privilege use. Audit CMU-authenticated sessions separately for clear forensic traceability.
☐	Break-glass DBA account	Keep at least one local DBA account with a vaulted password in your secrets manager. Test it. Document the runbook. Do this before go-live.

Phase 6: Testing and Go-Live

Validate before you flip the switch

Test with real AD users in a non-production environment that mirrors production. Access and privilege issues found here take minutes to fix. Found in production, they take hours.

	Task	What to verify
☐	Multi-role user testing	Test at least one user per AD group. Verify SESSION_ROLES matches the expected Oracle privileges for each role.
☐	Role inheritance	If AD groups have nested membership, verify Oracle resolves nested group-to-role mappings correctly.
☐	AD failover scenario	Confirm break-glass local account connects. Confirm that application error handling is acceptable.
☐	Login performance	CMU adds one LDAP round-trip per new session. Baseline login time before and after.
☐	Local account migration plan	Document which local accounts are deprecated, which stay, and the cutover timeline. Communicate to all affected users before go-live.
☐	Go/no-go sign-off	Written confirmation from DBA lead, AD admin, Security, and application owner. All four. Not verbal.

PRO TIP

Run a go-live rehearsal with a non-DBA user. If they need help logging in, your runbook and communication are not ready yet.

Quick Reference: CMU vs Local Oracle Users

Factor	CMU via Active Directory	Local Oracle Users
User provisioning	AD team handles via HR workflow or Identity Management (IDM) systems	DBA creates each account manually per database
Password policy	AD enforces globally. One policy, all databases.	Each database enforces independently via profiles
Offboarding	Disable AD account. Access revoked across all databases.	DBA must lock or drop account in every database
Role management	AD group membership drives Oracle role assignment	GRANT/REVOKE per user per database
Scales to many databases	Yes. No additional DBA work per new database.	Linear effort. Each database requires a manual setup.

Factor	CMU via Active Directory	Local Oracle Users
Password resets	AD help desk handles. Zero DBA involvement.	DBA resets per ALTER USER in each database.
Upfront complexity	One time Wallet, LDAP config, AD schema mapping.	Lower. Standard CREATE USER and GRANT. Complexity grows quickly.

Get CMU Right the First Time.

Schema strategy, role design, Active Directory integration, security reviews, and testing plans have a much larger impact on long-term success than the installation itself.

Planning a CMU Rollout?

Viscosity Oracle consultants help organizations design secure, scalable CMU deployments, from Active Directory integration and role design to production rollout planning.

Explore Oracle Professional Services:

<https://viscosityna.com/professional-services/oracle-database-26ai>