Oracle® Database

Database Client Installation Guide

23ai for Linux

F46810-12

April 2025

ORACLE®

Oracle Database Database Client Installation Guide, 23ai for Linux

Primary Authors: Prakash Jashnani, Tejas N K

Contributing Authors: Douglas Williams

Contributors: Subrahmanyam Kodavaluru, Srinivas Poovala, Angad Gokakkar, Sharad Chandran R, John Arackal, Mugilan M, Xiaoyu Zhou, Prasad Bagal, Subhranshu Banerjee, Tammy Bednar, Jean-Francois Verrier, Neha Avasthy, Ranjit Noronha, Linto John, Robert Chang, Darcy Christensen, Jonathan Creighton, Benoit Dageville, Sudip Datta, Jim Erickson, Marcus Fallen, Joseph Francis, Allan Graves, Barbara Glover, Thirumaleshwara Hasandka, Sergio Leunissen, Clara Jaeckel, Aneesh Khandelwal, Joel Kallman, Eugene Karichkin, Jai Krishnani, Sangeeth Kumar, Kevin Jernigan, Christopher Jones, Prasad Kuruvadi Nagaraj, Bryn Llewellyn, Saar Maoz, Sunil Surabhi, Gopal Mulagund, Sue Lee, Rich Long, Barb Lundhild, Rolly Lv, Rudregowda Mallegowda, Padmanabhan Manavazhi, Krishna Mohan, John McHugh, Gurudas Pai, Satish Panchumarthy , Rajesh Prasad, Rajendra Pingte, Mohammed Shahnawaz Quadri, Hanlin Qian, Hema Ramamurthy, Sunil Ravindrachar, Mark Richwine, Dipak Saggi, Shachi Sanklecha, Mohit Singhal, Dharma Sirnapalli, Akshay Shah, James Spiller, Roy Swonger, Binoy Sukumaran, Ravi Thammaiah, Shekhar Vaggu, Peter Wahl, Sergiusz Wolicki

Contents

ORACLE

4   Configuring Users, Groups and Environments for Oracle Database Client

5   Installing Oracle Database Client

ORACLE®

6   Installing Oracle Instant Client

7   Oracle Database Client Postinstallation Tasks

8   Removing Oracle Database Software

A   Response Files

Preface

This guide explains how to install and configure Oracle Database Client.

This guide also provides information about postinstallation tasks and how to remove the database client software.

Oracle Database Client enables development and deployment of applications that connect to Oracle Database, either on-premises or in the Cloud. The Oracle Database Client libraries provide the necessary network connectivity and advanced data features to make full use of Oracle Database. Tools, libraries, and SDKs included in Oracle Database Client provide quick and convenient data access.

- **Audience**
  This guide is intended for anyone responsible for installing Oracle Database Client 23ai.
- **Documentation Accessibility**
- **Diversity and Inclusion**
- **Set Up Java Access Bridge to Implement Java Accessibility**
  Install Java Access Bridge so that assistive technologies on Microsoft Windows systems can use the Java Accessibility API.
- **Command Syntax**
  Refer to these command syntax conventions to understand command examples in this guide.
- **Conventions**

Audience

This guide is intended for anyone responsible for installing Oracle Database Client 23ai.

Additional installation guides for Oracle Database, Oracle Real Application Clusters, Oracle Clusterware, Oracle Database Examples, and Oracle Enterprise Manager Cloud Control are available at the following URL:

http://docs.oracle.com

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Set Up Java Access Bridge to Implement Java Accessibility

Install Java Access Bridge so that assistive technologies on Microsoft Windows systems can use the Java Accessibility API.

Java Access Bridge is a technology that enables Java applications and applets that implement the Java Accessibility API to be visible to assistive technologies on Microsoft Windows systems.

Refer to *Java Platform, Standard Edition Accessibility Guide* for information about the minimum supported versions of assistive technologies required to use Java Access Bridge. Also refer to this guide to obtain installation and testing instructions, and instructions for how to use Java Access Bridge.

Related Topics

• *Java Platform, Standard Edition Java Accessibility Guide*

Command Syntax

Refer to these command syntax conventions to understand command examples in this guide.

Convention	Description
`$`	Bourne or BASH shell prompt in a command example. Do not enter the prompt as part of the command.
`%`	C Shell prompt in a command example. Do not enter the prompt as part of the command.
`#`	Superuser (root) prompt in a command example. Do not enter the prompt as part of the command.
`monospace`	UNIX command syntax
backslash \	A backslash is the UNIX and Linux command continuation character. It is used in command examples that are too long to fit on a single line. Enter the command as displayed (with a backslash) or enter it on a single line without a backslash: `dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b \  count=10000`
braces { }	Braces indicate required items: `.DEFINE {macro1}`

Convention	Description
brackets [ ]	Brackets indicate optional items: `cvtcrt termname [outfile]`
ellipses ...	Ellipses indicate an arbitrary number of similar items: `CHKVAL fieldname value1 value2 ... valueN`
italic	Italic type indicates a variable. Substitute a value for the variable: `library_name`
vertical line |	A vertical line indicates a choice within braces or brackets: `FILE filesize [K|M]`

Conventions

The following text conventions are used in this document:

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

1
Oracle Database Client Installation Checklist

Use checklists to review system requirements, and to plan and carry out Oracle Database Client installation.

Oracle recommends that you use checklists as part of your installation planning process. Using checklists can help you to confirm that your server hardware and configuration meet minimum requirements for this release and can help you carry out a successful installation.

- Server Hardware Checklist for Oracle Database Client Installation
  Use this checklist to check hardware requirements for Oracle Database Client installations.

- Operating System Checklist for Oracle Database Client on Linux
  Use this checklist to check minimum operating system requirements for Oracle Database Client.

- Server Configuration Checklist for Oracle Database Client
  Use this checklist to check minimum server configuration requirements for Oracle Database Client installations.

- Oracle User Environment Configuration Checklist for Oracle Database Installation
  Use this checklist to plan operating system users, groups, and environments for Oracle Database management.

- Storage Checklist for Oracle Database Client
  Use this checklist to review storage minimum requirements and assist with configuration planning.

- Installer Planning Checklist for Oracle Database Client
  Use this checklist to assist you to be prepared before starting Oracle Universal Installer.

Server Hardware Checklist for Oracle Database Client Installation

Use this checklist to check hardware requirements for Oracle Database Client installations.

Table 1-1   Server Hardware Checklist for Oracle Database Client Installations

Check	Task
Runlevel	3 or 5
Server Display Cards	At least 1024 x 768 display resolution, which Oracle Universal Installer requires.
Minimum network connectivity	Client is connected to a network.
Minimum RAM	At least 256 MB of RAM.
Linux for Arm (aarch64) processor	Oracle Database 23ai Linux for Arm (aarch64) requires a CPU capable of Neoverse N1. Oracle recommends using a server with a minimum of Ampere Altra or Ampere Altra Max CPU.

Operating System Checklist for Oracle Database Client on Linux

Use this checklist to check minimum operating system requirements for Oracle Database Client.

> **Note:**
>
> Oracle Database 23ai is currently not supported on non-Oracle hardware. For support, availability, and database offerings, see the *Oracle Database Licensing Information User Manual* and My Oracle Support Note 742060.1.

Table 1-2   Operating System General Checklist for Oracle Database Client on Linux

Item	Task
Operating system general requirements	OpenSSH installed manually, if you do not have it installed already as part of a default Linux installation.
	A Linux kernel in the list of supported kernels and releases listed in this guide.
Linux x86-64 operating system requirements	The following Linux x86-64 kernels are supported:
	• Oracle Linux 9.2 with the Unbreakable Enterprise Kernel 7: 5.15.0-201.135.6.el9uek.x86_64 or later Oracle Linux 9.2 with the Red Hat Compatible Kernel: 5.14.0-284.30.1.el9_2.x86_64 or later
	• Oracle Linux 8.8 with the Unbreakable Enterprise Kernel 7: 5.15.0-202.135.2.el8uek.x86_64 or later Oracle Linux 8.6 with the Unbreakable Enterprise Kernel 6: 5.4.17-2136.312.3.4.el8uek.x86_64 or later Oracle Linux 8.6 with the Red Hat Compatible Kernel: 4.18.0-372.26.1.0.1.el8_6.x86_64 or later
	• Red Hat Enterprise Linux 9.2: 5.14.0-284.30.1.el9_2.x86_64 or later
	• Red Hat Enterprise Linux 8.6: 4.18.0-372.26.1.0.1.el8_6.x86_64 or later
	• SUSE Linux Enterprise Server 15 SP5 (x86_64): 5.14.21-150500.53-default or later
	• Ubuntu 22.04.1 LTS 5.15.0-1016 or later **Note:** Oracle Database Client supports only 64-bit client installations on Ubuntu.
	• Debian GNU/Linux 11 (bullseye) 5.10.120-1 or later **Note:** Oracle Database Client supports only 64-bit client installations on Debian.
	Review the system requirements section for a list of minimum package requirements.
Linux for Arm (aarch64) operating system requirements	Starting with Oracle Database 23ai Release 23.5, the Linux for Arm (aarch64) operating system is supported with the following minimum supported version:
	• Oracle Linux 8.6 with Unbreakable Enterprise Kernel 7: 5.15.0-205.149.5.1.el8uek.aarch64 **Note:** Oracle recommends that you update Oracle Linux 8 to the latest available release level.

Table 1-2   (Cont.) Operating System General Checklist for Oracle Database Client on Linux

Item	Task
Oracle Preinstallation RPM for Oracle Linux	If you use Oracle Linux, then Oracle recommends that you run an Oracle preinstallation RPM for your Linux release to configure your operating system for Oracle Database and Oracle Grid Infrastructure installations.

Server Configuration Checklist for Oracle Database Client

Use this checklist to check minimum server configuration requirements for Oracle Database Client installations.

Table 1-3   Server Configuration Checklist for Oracle Database Client

Check	Task
Disk space allocated to the `/tmp` directory	At least 400 MB of space in the temporary disk space (`/tmp`) directory.
Swap space allocation relative to RAM	256 MB: 3 times the size of RAM Between 256 MB and 512 MB: 2 times the size of RAM Between 512 MB and 2 GB: 1.5 times the size of RAM Between 2 GB and 16 GB: Equal to the size of RAM More than 16 GB: 16 GB **Note:** If you enable HugePages for your Linux servers, then you should deduct the memory allocated to HugePages from the available RAM before calculating swap space.
Oracle Inventory (oraInventory) and OINSTALL Group Requirements	• For new installs, if you have not configured an oraInventory directory, then the installer creates an Oracle inventory that is one directory level up from the Oracle base for the Oracle Grid Infrastructure install, and designates the installation owner's primary group as the Oracle Inventory group. • For upgrades, Oracle Universal Installer (OUI) detects an existing oraInventory directory from the `/etc/oraInst.loc` file, and uses the existing oraInventory. The Oracle Inventory directory is the central inventory of Oracle software installed on your system. Users who have the Oracle Inventory group as their primary group are granted the OINSTALL privilege to write to the central inventory. The OINSTALL group must be the primary group of all Oracle software installation owners on the server. It should be writable by any Oracle installation owner.
Groups and users	Oracle recommends that you create groups and user accounts required for your security plans before starting installation. Installation owners have resource limits settings and other requirements. Group and user names must use only ASCII characters.
Mount point paths for the software binaries	Oracle recommends that you create an Optimal Flexible Architecture configuration as described in the appendix "Optimal Flexible Architecture" in *Oracle Database Installation Guide* for your platform.
Ensure that the Oracle home (the Oracle home path you select for Oracle Database) uses only ASCII characters	The ASCII character restriction includes installation owner user names, which are used as a default for some home paths, as well as other directory names you may select for paths.

Table 1-3   (Cont.) Server Configuration Checklist for Oracle Database Client

Check	Task
Determine `root` privilege delegation option for installation	During installation, you are asked to run configuration scripts as the `root` user. You can either run these scripts manually as `root` when prompted, or you can provide configuration information and passwords using a root privilege delegation option such as Sudo.
	To enable Sudo, have a system administrator with the appropriate privileges configure a user that is a member of the sudoers list, and provide the username and password when prompted during installation.
Set locale (if needed)	Specify the language and the territory, or locale, in which you want to use Oracle components. A locale is a linguistic and cultural environment in which a system or program is running. NLS (National Language Support) parameters determine the locale-specific behavior on both servers and clients. The locale setting of a component determines the language of the user interface of the component, and the globalization behavior, such as date and number formatting.
symlinks	Oracle home or Oracle base cannot be `symlinks`, nor can any of their parent directories, all the way to up to the `root` directory.

Related Topics

*   *Oracle Database Globalization Support Guide*

Oracle User Environment Configuration Checklist for Oracle Database Installation

Use this checklist to plan operating system users, groups, and environments for Oracle Database management.

Table 1-4   User Environment Configuration for Oracle Database

Check	Task
Review Oracle Inventory (oraInventory) and OINSTALL Group Requirements	The physical group you designate as the Oracle Inventory directory is the central inventory of Oracle software installed on your system. It should be the primary group for all Oracle software installation owners. Users who have the Oracle Inventory group as their primary group are granted the OINSTALL privilege to read and write to the central inventory.
	• If you have an existing installation, then OUI detects the existing oraInventory directory from the `/etc/oraInst.loc` file, and uses this location.
	• If you are installing Oracle software for the first time, then you can specify the Oracle inventory directory and the Oracle base directory during the Oracle software installation, and Oracle Universal Installer will set up the software directories for you. Ensure that the directory paths that you specify are in compliance with the Oracle Optimal Flexible Architecture recommendations.
	Ensure that the group designated as the OINSTALL group is available as the primary group for all planned Oracle software installation owners.

Table 1-4   (Cont.) User Environment Configuration for Oracle Database

Check	Task
Create operating system groups and users for standard or role-allocated system privileges	The Preinstallation RPM automatically creates the `oracle` user and the oraInventory (`oinstall`) and OSDBA (`dba`) groups for you.
	Create additional operating system groups and users depending on your security requirements as described in this installation guide.
	Set resource limits settings and other requirements for Oracle software installation owners.
	Group and user names must use only ASCII characters.
Unset Oracle Software Environment Variables	If you have had an existing installation on your system, and you are using the same user account for this installation, then unset the ORACLE_HOME, ORACLE_BASE, ORACLE_SID, TNS_ADMIN environment variables and any other environment variable set for the Oracle installation user that is connected with Oracle software homes.
Configure the Oracle Software Owner Environment	Configure the environment of the `oracle` or `grid` user by performing the following tasks:
	• Set the default file mode creation mask (`umask`) to 022 in the shell startup file.
	• Set the `DISPLAY` environment variable.

Storage Checklist for Oracle Database Client

Use this checklist to review storage minimum requirements and assist with configuration planning.

Table 1-5   Storage Checklist for Oracle Database Client

Check	Task
Minimum local disk storage space for Oracle Database Client software	For Linux x86-64: At least 2.3 GB for Administrator installation type. At least 1.9 GB for Runtime installation type. At least 2.3 GB for Custom installation type. At least 2.4 GB for image-based client installation. For Linux for Arm (aarch64): At least 1.9 GB for Administrator installation type. At least 1.5 GB for Runtime installation type. At least 1.9 GB for Custom installation type. At least 1.8 GB for image-based client installation.

Installer Planning Checklist for Oracle Database Client

Use this checklist to assist you to be prepared before starting Oracle Universal Installer.

ORACLE

Table 1-6   Oracle Universal Installer Planning Checklist for Oracle Database Client Installation

Check	Task
Read the Release Notes	Review release notes for your platform, which are available for your release at the following URL: http://docs.oracle.com/en/database/database.html
Review the Licensing Information	You are permitted to use only those components in the Oracle Database media pack for which you have purchased licenses. For more information about licenses, refer to the following URL: *Oracle Database Licensing Information*
Review Oracle Support Certification Matrix	New platforms and operating system software versions might be certified after this guide is published, review the certification matrix on the My Oracle Support website for the most up-to-date list of certified hardware platforms and operating system versions: https://support.oracle.com/ You must register online before using My Oracle Support. After logging in, from the menu options, select the **Certifications** tab. On the Certifications page, use the **Certification Search** options to search by Product, Release, and Platform. You can also search using the **Certification Quick Link** options such as Product Delivery, and Lifetime Support.
Run OUI with CVU and use fixup scripts	Oracle Universal Installer is fully integrated with Cluster Verification Utility (CVU), automating many CVU prerequisite checks. Oracle Universal Installer runs all prerequisite checks and creates fixup scripts when you run the installer. You can run OUI up to the Summary screen without starting the installation. You can also run CVU commands manually to check system readiness. For more information, see: *Oracle Clusterware Administration and Deployment Guide*
Ensure `cron` jobs do not run during installation	If the installer is running when daily cron jobs start, then you may encounter unexplained installation problems if your cron job is performing cleanup, and temporary files are deleted before the installation is finished. Oracle recommends that you complete installation before daily cron jobs are run, or disable daily cron jobs that perform cleanup until after the installation is completed.
Decide the client installation type	You can choose one of the following installation types when installing Oracle Database Client using the Setup Wizard: • **Administrator:** Enables applications to connect to an Oracle Database instance on the local system or on a remote system. It also provides tools that enable you to administer Oracle Database. • **Runtime:** Enables applications to connect to an Oracle Database instance on the local system or on a remote system. • **Custom:** Enables you to select individual components from the list of Administrator and Runtime components. **Instant Client:** Enables you to install only the shared libraries required by Oracle Call Interface (OCI), Oracle C++ Call Interface (OCCI), Pro*C, or Java database connectivity (JDBC) OCI applications. This installation type requires much less disk space than the other Oracle Database Client installation types. For more information about Oracle Database Instant Client see the following URL: https://www.oracle.com/database/technologies/instant-client.html **Note:** Starting with Oracle Database Client 23ai, the Setup Wizard no longer provides the option to install Oracle Instant Client.

Table 1-6   (Cont.) Oracle Universal Installer Planning Checklist for Oracle Database Client Installation

Check	Task
Obtain your My Oracle Support account information.	During installation, you require a My Oracle Support user name and password to configure security updates, download software updates, and other installation tasks. You can register for My Oracle Support at the following URL:
	https://support.oracle.com/
Oracle Database Client and Oracle Database interoperability	• For information about interoperability between Oracle Database Client and Oracle Database releases, see My Oracle Support Note 207303.1. • 32-bit Oracle Database clients are desupported in Oracle Database 23ai.
	Oracle has discontinued developing 32-bit Oracle Database clients. Oracle recommends that you use 64-bit Oracle Database clients. If you are using 32-bit applications, then you can continue to use older 32-bit Oracle Database clients, subject to "Client / Server Interoperability Support Matrix for Different Oracle Versions (Doc ID 207303.1)," and the release support lifecycle (My Oracle Support 742060.1).

2

Checking and Configuring Server Hardware for Oracle Database Client

Verify that servers where you install Oracle Database Client meet the minimum requirements for installation.

This section provides minimum server requirements to complete installation of Oracle Database Client. It does not provide system resource guidelines, or other tuning guidelines for particular workloads.

- Logging In to a Remote System Using X Window System
  Use this procedure to run Oracle Universal Installer (OUI) by logging on to a remote system where the runtime setting prohibits logging in directly to a graphical user interface (GUI).

- Checking Server Hardware and Memory Configuration
  Use this procedure to gather information about your server configuration.

Logging In to a Remote System Using X Window System

Use this procedure to run Oracle Universal Installer (OUI) by logging on to a remote system where the runtime setting prohibits logging in directly to a graphical user interface (GUI).

OUI is a graphical user interface (GUI) application. On servers where the runtime settings prevent GUI applications from running, you can redirect the GUI display to a client system connecting to the server.

> ✏️ **Note:**
>
> If you log in as another user (for example, `oracle` or `grid`), then repeat this procedure for that user as well.

1. Start an X Window System session. If you are using an X Window System terminal emulator from a PC or similar system, then you may need to configure security settings to permit remote hosts to display X applications on your local system.

2. Enter a command using the following syntax to enable remote hosts to display X applications on the local X server:

```
# xhost + RemoteHost
```

RemoteHost is the fully qualified remote host name. For example:

```
# xhost + somehost.example.com
somehost.example.com being added to the access control list
```

3. If you are not installing the software on the local system, then use the `ssh` command to connect to the system where you want to install the software:

```
# ssh -Y  RemoteHost
```

RemoteHost is the fully qualified remote host name. The `-Y` flag ("yes") enables remote clients to have full access to the original server display. For example:

```
# ssh -Y somehost.example.com
```

4. If you are not logged in as the `root` user, and you are performing configuration steps that require `root` user privileges, then switch the user to `root`.

> **Note:**
>
> For more information about remote login using X Window System, refer to your X server documentation, or contact your X server vendor or system administrator. Depending on the X server software that you are using, you may have to complete the tasks in a different order.

Checking Server Hardware and Memory Configuration

Use this procedure to gather information about your server configuration.

1. Use the following command to determine physical RAM size on the server:

```
# grep MemTotal /proc/meminfo
```

If the size of the physical RAM installed in the system is less than the required size, then you must install more memory before continuing.

2. Determine the size of the configured swap space:

```
# grep SwapTotal /proc/meminfo
```

If necessary, see your operating system documentation for information about how to configure additional swap space.

3. Determine the amount of space available in the `/tmp` directory:

```
# df -h /tmp
```

If the free space available in the `/tmp` directory is less than what is required, then complete one of the following steps:

- Delete unused files from the `/tmp` directory to meet the disk space requirement.

> **✎ Note:**
>
> If you perform this step after installing Oracle software, then do not
> remove `/tmp/.oracle` or `/var/tmp/.oracle` directories or their files.

- When you set the Oracle user's environment, also set the `TMP` and `TMPDIR` environment variables to the directory you want to use instead of `/tmp`.

4. Determine the amount of free RAM and disk swap space on the system:

```
# free
```

5. Determine if the system architecture can run the software:

```
# uname -m
```

Verify that the processor architecture matches the Oracle software release to install. For example, you should see the following for a x86-64 bit system:

```
x86_64
```

If you do not see the expected output, then you cannot install the software on this system.

6. Verify that shared memory (`/dev/shm`) is mounted properly with sufficient size:

```
df -h /dev/shm
```

The `df-h` command displays the filesystem on which `/dev/shm` is mounted, and also displays in GB the total size and free size of shared memory.

3

Configuring Operating Systems for Oracle Database Client on Linux

Complete operating system configuration requirements and checks for Linux operating systems before you start installation.

- **About Oracle Linux with the Unbreakable Enterprise Kernel**
  The Unbreakable Enterprise Kernel for Oracle Linux provides the latest innovations from upstream development to customers who run Oracle Linux in the data center.

- **Reviewing Operating System Security Common Practices**
  Secure operating systems are an important basis for general system security.

- **About Operating System Requirements**
  Depending on the products that you intend to install, verify that you have the required operating system kernel and packages installed.

- **Operating System Requirements for x86-64 Linux Platforms**
  The Linux distributions and packages listed in this section are supported for this release on x86-64.

- **Operating System Requirements for Linux for Arm (aarch64)**
  The Linux distribution and packages listed in this section are supported for this release on Linux for Arm (aarch64).

- **Additional Drivers and Software Packages for Linux**
  Information about optional drivers and software packages.

- **Checking Kernel and Package Requirements for Linux**
  Verify your kernel and packages to see if they meet minimum requirements for installation.

- **Verifying Digital Signature and Integrity of Installation Archive Files**
  Starting with Oracle Database 23ai, Oracle digitally signs the installation archive files with Oracle certificates to ensure the integrity of the packages before you deploy them in your environments.

About Oracle Linux with the Unbreakable Enterprise Kernel

The Unbreakable Enterprise Kernel for Oracle Linux provides the latest innovations from upstream development to customers who run Oracle Linux in the data center.

The Unbreakable Enterprise Kernel is included and enabled by default in Oracle Linux kernels. It is based on a recent stable mainline development Linux kernel, and also includes optimizations developed in collaboration with Oracle Database, Oracle middleware, and Oracle hardware engineering teams to ensure stability and optimal performance for the most demanding enterprise workloads.

Oracle highly recommends deploying the Unbreakable Enterprise Kernel in your Oracle Linux environment, especially if you run enterprise applications. However, using Unbreakable Enterprise Kernel is optional. If you require strict Red Hat Enterprise Linux (RHEL) kernel compatibility, then Oracle Linux also includes a kernel compatible with the RHEL Linux kernel, compiled directly from the RHEL source code.

You can obtain more information about the Unbreakable Enterprise Kernel for Oracle Linux at the following URL:

https://www.oracle.com/linux/

The Unbreakable Enterprise Kernel for Oracle Linux is the standard kernel used with Oracle products. The build and QA systems for Oracle Database and other Oracle products use the Unbreakable Enterprise Kernel for Oracle Linux exclusively. The Unbreakable Enterprise Kernel for Oracle Linux is also the kernel used in Oracle Exadata systems. Unbreakable Enterprise Kernel for Oracle Linux is used in all benchmark tests on Linux in which Oracle participates, as well as in the Oracle Database Preinstallation RPM program for x86-64.

Oracle Ksplice, which is part of Oracle Linux, updates the Linux operating system (OS) kernel, while it is running, without requiring restarts or any interruption. Ksplice is available only with Oracle Linux.

Reviewing Operating System Security Common Practices

Secure operating systems are an important basis for general system security.

Ensure that your operating system deployment is in compliance with common security practices as described in your operating system vendor security guide.

About Operating System Requirements

Depending on the products that you intend to install, verify that you have the required operating system kernel and packages installed.

Requirements listed in this document are current as of the date listed on the title page.

Oracle Universal Installer performs checks on your system to verify that it meets the listed operating system package requirements. To ensure that these checks complete successfully, verify the requirements before you start OUI.

> **Note:**
>
> Oracle does not support running different operating system versions on cluster members, unless an operating system is being upgraded. You cannot run different operating system version binaries on members of the same cluster, even if each operating system is supported.

Operating System Requirements for x86-64 Linux Platforms

The Linux distributions and packages listed in this section are supported for this release on x86-64.

Identify the requirements for your Linux distribution, and ensure that you have a supported kernel and required packages installed before starting installation.

The platform-specific hardware and software requirements included in this guide were current when this guide was published. However, because new platforms and operating system software versions may be certified after this guide is published, review the certification matrix on the My Oracle Support website for the most up-to-date list of certified hardware platforms and operating system versions:

https://support.oracle.com/

> **Note:**
>
> Oracle Database 23ai is currently not supported on non-Oracle hardware. For
> support, availability, and database offerings, see the *Oracle Database Licensing
> Information User Manual* and My Oracle Support Note 742060.1.

- **General Guidelines for Operating System Packages**
  Review these guidelines before you proceed with installing operating system packages.

- **Supported Oracle Linux 9 Distributions for x86-64**
  Use the following information to check supported Oracle Linux 9 distributions:

- **Supported Oracle Linux 8 Distributions for x86-64**
  Use the following information to check supported Oracle Linux 8 distributions:

- **Supported Red Hat Enterprise Linux 9 Distributions for x86-64**
  Use the following information to check supported Red Hat Enterprise Linux 9 distributions:

- **Supported Red Hat Enterprise Linux 8 Distributions for x86-64**
  Use the following information to check supported Red Hat Enterprise Linux 8 distributions:

- **Supported SUSE Linux Enterprise Server 15 Distributions for x86-64**
  Use the following information to check supported SUSE Linux Enterprise Server 15
  distributions:

- **Supported Ubuntu Distributions for x86-64**
  Use the following information to check supported Ubuntu distributions:

- **Supported Debian Distributions for x86-64**
  Use the following information to check supported Debian GNU/Linux 10 distributions:

- **Installing Operating System Packages**
  Learn how to install the latest version of your Oracle Linux and SUSE Linux Enterprise
  Server operating system packages listed earlier.

General Guidelines for Operating System Packages

Review these guidelines before you proceed with installing operating system packages.

- Oracle recommends that you install Oracle Linux and use Oracle Database Preinstallation
  RPM to configure your operating systems for Oracle Database and Oracle Grid
  Infrastructure installations.

- The Oracle Database interactive installation requires an X Window System (for example,
  `libX`). The `libX` packages are not part of a default Linux installation. If you perform an
  installation on a system with a reduced set of packages, then you must ensure that `libX` or
  a similar X Window System package is installed.

- Oracle Database 12c Release 2 (12.2) and later does not require the compiler packages
  `gcc` and `gcc-c++` on for Oracle Database or Oracle Grid Infrastructure installations.

- These operating system requirements do not apply to Oracle Engineered Systems, such
  as Oracle Exadata Database Machine and Oracle Database Appliance. Oracle Engineered
  Systems include integrated system software that contain the required version of the
  operating system kernel and all software packages. Please verify that you have the
  minimum required Exadata image. Refer My Oracle Support note 888828.1 and 888888.1
  for more information.

Related Topics

- My Oracle Support Note 888828.1
- My Oracle Support Note 888888.1

Supported Oracle Linux 9 Distributions for x86-64

Use the following information to check supported Oracle Linux 9 distributions:

Table 3-1   x86-64 Oracle Linux 9 Minimum Operating System Requirements

Item	Requirements
SSH Requirement	Ensure that OpenSSH is installed on your servers. OpenSSH is the required SSH software.
Oracle Linux 9	Minimum supported versions: • Oracle Linux 9.2 with the Unbreakable Enterprise Kernel 7: 5.15.0-201.135.6.el9uek.x86_64 or later • Oracle Linux 9.2 with the Red Hat Compatible Kernel: 5.14.0-284.30.1.el9_2.x86_64 or later **Note:** Oracle recommends that you update Oracle Linux to the latest available version and release level.

Table 3-1   (Cont.) x86-64 Oracle Linux 9 Minimum Operating System Requirements

Item	Requirements
Packages for Oracle Linux 9	Subscribe to the Oracle Linux 9 channel on the Unbreakable Linux Network, or configure a yum repository from the Oracle Linux yum server website, and then install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`. The Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, automatically installs all required packages listed in the table below, their dependencies for Oracle Grid Infrastructure and Oracle Database installations, and also performs other system configuration. If you install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, then you do not have to install these packages, as the Oracle Database Preinstallation RPM automatically installs them. bc binutils compat-openssl11 elfutils-libelf fontconfig glibc glibc-devel glibc-headers ksh libaio libasan liblsan libX11 libXau libXi libXrender libXtst libxcrypt-compat libgcc libibverbs librdmacm libstdc++ libxcb libvirt-libs make policycoreutils policycoreutils-python-utils smartmontools sysstat
Optional Packages for Oracle Linux 9	Based on your requirement, install the latest released versions of the following packages: ipmiutil (for Intelligent Platform Management Interface) libnsl2 (for Oracle Database Client only) libnsl2-devel (for Oracle Database Client only) net-tools (for Oracle RAC and Oracle Clusterware) nfs-utils (for Oracle ACFS)

Supported Oracle Linux 8 Distributions for x86-64

Use the following information to check supported Oracle Linux 8 distributions:

Table 3-2   x86-64 Oracle Linux 8 Minimum Operating System Requirements

Item	Requirements
SSH Requirement	Ensure that OpenSSH is installed on your servers. OpenSSH is the required SSH software.
Oracle Linux 8	Minimum supported versions: • Oracle Linux 8.6 with the Unbreakable Enterprise Kernel 6: 5.4.17-2136.312.3.4.el8uek.x86_64 or later • Oracle Linux 8.6 with the Red Hat Compatible Kernel: 4.18.0-372.26.1.0.1.el8_6.x86_64 or later Starting with Oracle Database 23ai release 23.5, the following minimum version is supported: • Oracle Linux 8.8 with the Unbreakable Enterprise Kernel 7: 5.15.0-202.135.2.el8uek.x86_64 or later **Note:** Oracle recommends that you update Oracle Linux to the latest available version and release level.

Table 3-2   (Cont.) x86-64 Oracle Linux 8 Minimum Operating System Requirements

Item	Requirements
Packages for Oracle Linux 8	Subscribe to the Oracle Linux 8 channel on the Unbreakable Linux Network, or configure a yum repository from the Oracle Linux yum server website, and then install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`. The Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, automatically installs all required packages listed in the table below, their dependencies for Oracle Grid Infrastructure and Oracle Database installations, and also performs other system configuration. If you install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, then you do not have to install these packages, as the Oracle Database Preinstallation RPM automatically installs them. bc binutils compat-openssl10 elfutils-libelf fontconfig glibc glibc-devel ksh libaio libXrender libX11 libXau libXi libXtst libgcc libstdc++ libxcb libibverbs libasan liblsan librdmacm make policycoreutils policycoreutils-python-utils smartmontools sysstat
Optional Packages for Oracle Linux 8	Based on your requirement, install the latest released versions of the following packages: ipmiutil (for Intelligent Platform Management Interface) libnsl2 (for Oracle Database Client only) libnsl2-devel (for Oracle Database Client only) net-tools (for Oracle RAC and Oracle Clusterware) nfs-utils (for Oracle ACFS)

Supported Red Hat Enterprise Linux 9 Distributions for x86-64

Use the following information to check supported Red Hat Enterprise Linux 9 distributions:

Table 3-3   x86-64 Red Hat Enterprise Linux 9 Minimum Operating System Requirements

Item	Requirements
SSH Requirement	Ensure that OpenSSH is installed on your servers. OpenSSH is the required SSH software.
Red Hat Enterprise Linux 9	Minimum supported version: Red Hat Enterprise Linux 9.2: 5.14.0-284.30.1.el9_2.x86_64 or later
Packages for Red Hat Enterprise Linux 9	Install the latest released versions of the following packages: bc binutils compat-openssl11 elfutils-libelf fontconfig glibc glibc-devel glibc-headers ksh libaio libasan liblsan libX11 libXau libXi libXrender libXtst libxcrypt-compat libgcc libibverbs librdmacm libstdc++ libxcb libvirt-libs make policycoreutils policycoreutils-python-utils smartmontools sysstat
Optional Packages for Red Hat Enterprise Linux 9	Based on your requirement, install the latest released versions of the following packages: ipmiutil (for Intelligent Platform Management Interface) libnsl2 (for Oracle Database Client only) libnsl2-devel (for Oracle Database Client only) net-tools (for Oracle RAC and Oracle Clusterware) nfs-utils (for Oracle ACFS)

ORACLE

Supported Red Hat Enterprise Linux 8 Distributions for x86-64

Use the following information to check supported Red Hat Enterprise Linux 8 distributions:

Table 3-4   x86-64 Red Hat Enterprise Linux 8 Minimum Operating System Requirements

Item	Requirements
SSH Requirement	Ensure that OpenSSH is installed on your servers. OpenSSH is the required SSH software.
Red Hat Enterprise Linux 8	Minimum supported version: Red Hat Enterprise Linux 8.6: 4.18.0-372.26.1.0.1.el8_6.x86_64 or later
Packages for Red Hat Enterprise Linux 8	Install the latest released versions of the following packages: bc binutils compat-openssl10 elfutils-libelf fontconfig glibc glibc-devel ksh libaio libXrender libX11 libXau libXi libXtst libgcc libstdc++ libxcb libibverbs libasan liblsan librdmacm make policycoreutils policycoreutils-python-utils smartmontools sysstat
Optional Packages for Red Hat Enterprise Linux 8	Based on your requirement, install the latest released versions of the following packages: ipmiutil (for Intelligent Platform Management Interface) libnsl2 (for Oracle Database Client only) libnsl2-devel (for Oracle Database Client only) net-tools (for Oracle RAC and Oracle Clusterware) nfs-utils (for Oracle ACFS)

Supported SUSE Linux Enterprise Server 15 Distributions for x86-64

Use the following information to check supported SUSE Linux Enterprise Server 15 distributions:

Table 3-5   x86-64 SUSE Linux Enterprise Server 15 Minimum Operating System Requirements

Item	Requirements
SSH Requirement	Ensure that OpenSSH is installed on your servers. OpenSSH is the required SSH software.
SUSE Linux Enterprise Server	Minimum supported version: SUSE Linux Enterprise Server 15 SP5 (x86_64): 5.14.21-150500.53-default or later
Packages for SUSE Linux Enterprise Server 15	Install the latest released versions of the following packages: bc glibc gcc8 insserv-compat libaio-devel libaio1 libX11-6 libXau6 libXext-devel libXext6 libXi-devel libXi6 libXrender-devel libXrender1 libXtst6 libcap-ng-utils libcap-ng0 libcap-progs libcap1 libcap2 libelf1 libgcc_s1 libjpeg8 libpcap1 libpcre1 libpng16 libstdc++6 libgfortran4 mksh make pixz smartmontools sysstat xorg-x11-libs

Table 3-5   (Cont.) x86-64 SUSE Linux Enterprise Server 15 Minimum Operating System Requirements

Item	Requirements
Optional Packages for SUSE Linux Enterprise Server 15	Based on your requirement, install the latest released versions of the following packages: ipmiutil (for Intelligent Platform Management Interface) net-tools (for Oracle RAC and Oracle Clusterware)

Supported Ubuntu Distributions for x86-64

Use the following information to check supported Ubuntu distributions:

Table 3-6   Ubuntu Minimum Operating System Requirements

Item	Requirements
SSH Requirement	Ensure that OpenSSH is installed on your servers. OpenSSH is the required SSH software.
Ubuntu	Minimum supported version: Ubuntu 22.04.1 LTS 5.15.0-1016 or later **Note:** Oracle Database Client supports only 64-bit client installations on Ubuntu.
Packages for Ubuntu	Install the latest released versions of the following packages: bc binutils libaio1 libc6 libcap2 libgcc-s1 libpcap0.8 libstdc++6 libxi6 libxrender-dev libxtst6 make sysstat unzip x11-utils

Supported Debian Distributions for x86-64

Use the following information to check supported Debian GNU/Linux 10 distributions:

Table 3-7   x86-64 Debian GNU/Linux 10 Minimum Operating System Requirements

Item	Requirements
SSH Requirement	Ensure that OpenSSH is installed on your servers. OpenSSH is the required SSH software.
Debian GNU/Linux	Minimum supported version: • Debian GNU/Linux 11 (bullseye) 5.10.120-1 or later **Note:** Oracle Database Client supports only 64-bit client installations on Debian.
Packages for Debian GNU/Linux 10	Install the latest released versions of the following packages: binutils bc libaio1 libc6 libcap2 libgcc-s1 libpcap0.8 libstdc++6 libxi6 libxrender-dev libxtst6 make sysstat unzip x11-utils

Installing Operating System Packages

Learn how to install the latest version of your Oracle Linux and SUSE Linux Enterprise Server operating system packages listed earlier.

You must install the latest version of your operating system packages from the respective operating system vendor repository by using a package manager like `DNF`, `YUM` or `YaST` depending on your operating system.

Ensure that the appropriate channel or repository is enabled to include these packages.

For example:

On Oracle Linux or Red Hat Enterprise Linux, to install the latest `bc` package, run one of the following commands:

*   `# dnf install bc`

    (or)

*   `# yum install bc`

On SUSE Linux Enterprise Server, to install the latest `bc` package using `YaST`, run the following command:

```
$ yast --install bc
```

Operating System Requirements for Linux for Arm (aarch64)

The Linux distribution and packages listed in this section are supported for this release on Linux for Arm (aarch64).

Identify the requirements for your Linux distribution, and ensure that you have a supported kernel and required packages installed before starting installation.

The platform-specific hardware and software requirements included in this guide were current when this guide was published. However, because new platforms and operating system software versions may be certified after this guide is published, review the certification matrix on the My Oracle Support website for the most up-to-date list of certified hardware platforms and operating system versions:

https://support.oracle.com/

> **Note:**
>
> Oracle Database 23ai is currently not supported on non-Oracle hardware. For support, availability, and database offerings, see the *Oracle Database Licensing Information User Manual* and My Oracle Support Note 742060.1.

- Supported Oracle Linux 8 Distributions on Linux for Arm (aarch64)
  Use the following information to check supported Oracle Linux 8 distributions:

Supported Oracle Linux 8 Distributions on Linux for Arm (aarch64)

Use the following information to check supported Oracle Linux 8 distributions:

Table 3-8   Linux for Arm (aarch64) Oracle Linux 8 Minimum Operating System Requirements

Item	Requirements
Oracle Linux 8	Starting with Oracle Database 23ai Release 23.5, the Linux for Arm (aarch64) operating system is supported with the following minimum supported version:
	Oracle Linux 8.6 with Unbreakable Enterprise Kernel 7: 5.15.0-205.149.5.1.el8uek.aarch64
	Note: Oracle recommends that you update Oracle Linux 8 to the latest available release level.

Table 3-8 (Cont.) Linux for Arm (aarch64) Oracle Linux 8 Minimum Operating System Requirements

Item	Requirements
Packages for Oracle Linux 8	Install the latest released versions of the following packages:
	Subscribe to the Oracle Linux 8 channel on the Unbreakable Linux Network, or configure a yum repository from the Oracle Linux yum server website, and then install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`. The Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, automatically installs all required packages listed in the table below, their dependencies for Oracle Grid Infrastructure and Oracle Database installations, and also performs other system configuration. If you install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, then you do not have to install these packages, as the Oracle Database Preinstallation RPM automatically installs them.
	bc binutils elfutils-libelf gcc gcc-c++ glibc glibc-devel ksh libaio libaio-devel libgcc libgfortran libibverbs libnsl libnsl2 libstdc++ libstdc++-devel libxcb libX11 libXau libXi libXrender libXtst make policycoreutils policycoreutils-python-utils smartmontools sysstat

Additional Drivers and Software Packages for Linux

Information about optional drivers and software packages.

You are not required to install additional drivers and packages, but you may choose to install or configure these drivers and packages.

- Installing PAM for Login Authentication on Linux
  Pluggable Authentication Modules (PAM) is a system of libraries that handle user
  authentication tasks for applications.

- Installing Oracle Messaging Gateway
  Oracle Messaging Gateway is installed with Enterprise Edition of Oracle Database.
  However, you may require a CSD or Fix Packs.

- Installation Requirements for ODBC and LDAP
  Review these topics to install Open Database Connectivity (ODBC) and Lightweight
  Directory Access Protocol (LDAP).

- Installation Requirements for Programming Environments for Linux
  Review the following section to install programming environments:

- Installation Requirements for Web Browsers
  Web browsers are required only if you intend to use Oracle Enterprise Manager Cloud
  Control. Web browsers must support JavaScript, and the HTML 4.0 and CSS 1.0
  standards.

Installing PAM for Login Authentication on Linux

Pluggable Authentication Modules (PAM) is a system of libraries that handle user
authentication tasks for applications.

On Linux, external scheduler jobs require PAM. Oracle strongly recommends that you install
the latest Linux-PAM library for your Linux distribution.

Use a package management system (`yum`, `up2date`, `YaST`) for your distribution to install the
latest `pam` (Pluggable Authentication Modules for Linux) library.

Installing Oracle Messaging Gateway

Oracle Messaging Gateway is installed with Enterprise Edition of Oracle Database. However,
you may require a CSD or Fix Packs.

If you require a CSD or Fix Packs for IBM WebSphere MQ, then see the following website for
more information:

https://www.ibm.com/support/

> **Note:**
>
> - Oracle Messaging Gateway does not support the integration of Advanced
>   Queuing with TIBCO Rendezvous on IBM: Linux on System z.
>
> - Oracle Messaging Gateway is not supported on Linux for Arm (aarch64).

Related Topics

- *Oracle Database Advanced Queuing User's Guide*

Installation Requirements for ODBC and LDAP

Review these topics to install Open Database Connectivity (ODBC) and Lightweight Directory
Access Protocol (LDAP).

- **About ODBC Drivers and Oracle Database**
  Open Database Connectivity (ODBC) is a set of database access APIs that connect to the database, prepare, and then run SQL statements on the database.

- **Installing ODBC Drivers for Linux x86-64**
  If you intend to install Oracle's ODBC driver, then you must also install the most recent ODBC Driver Manager for Linux.

- **About LDAP and Oracle Plug-ins**
  Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing and maintaining distributed directory information services over IP networks.

- **Installing the LDAP Package**
  LDAP is included in a default Linux operating system installation.

About ODBC Drivers and Oracle Database

Open Database Connectivity (ODBC) is a set of database access APIs that connect to the database, prepare, and then run SQL statements on the database.

An application that uses an ODBC driver can access non-uniform data sources, such as spreadsheets and comma-delimited files.

Installing ODBC Drivers for Linux x86-64

If you intend to install Oracle's ODBC driver, then you must also install the most recent ODBC Driver Manager for Linux.

Download and install the ODBC Driver Manager and Linux RPMs from the following website:

http://www.unixodbc.org

Review the minimum supported ODBC driver releases, and install ODBC drivers of the following or later releases for all Linux distributions:

```
unixODBC-2.3.4 or later
```

About LDAP and Oracle Plug-ins

Lightweight Directory Access Protocol (LDAP) is an application protocol for accessing and maintaining distributed directory information services over IP networks.

You require the LDAP package if you want to use features requiring LDAP, including the Oracle Database scripts `odisrvreg` and `oidca` for Oracle Internet Directory, or `schemasync` for third-party LDAP directories.

Installing the LDAP Package

LDAP is included in a default Linux operating system installation.

If you did not perform a default Linux installation, and you intend to use Oracle scripts requiring LDAP, then use a package management system (`up2date`, `YaST`) for your distribution to install a supported LDAP package for your distribution, and install any other required packages for that LDAP package.

Installation Requirements for Programming Environments for Linux

Review the following section to install programming environments:

The Oracle JDBC Oracle Call Interface (OCI) Type 2 client driver (also known as a "thick" driver) is deprecated in Oracle Database 23ai.

Most Java applications that use Open Database Connectivity (ODBC) with Oracle JDBC drivers use the Thin driver. To enable Oracle to allocate resources to better address customer requirements, Oracle is deprecating the JDBC-OCI driver.

- Installation Requirements for Programming Environments for Linux x86-64
  Ensure that your system meets the requirements for the programming environment you want to configure:

- Installation Requirements for Programming Environments for Linux for Arm (aarch64)
  Ensure that your system meets the requirements for the programming environment you want to configure:

Installation Requirements for Programming Environments for Linux x86-64

Ensure that your system meets the requirements for the programming environment you want to configure:

Table 3-9   Requirements for Programming Environments for Linux X86–64

Programming Environments	Support Requirements
Java Database Connectivity (JDBC) / JDBC Oracle Call Interface (JDBC OCI)	JDK 11 (Java SE Development Kit) with the JNDI extension with Oracle Java Database Connectivity.
Oracle Call Interface (OCI) Oracle C++ Call Interface Pro*C/C++	Intel(R) C Intel(R) 64 Compiler Classic for applications running on Intel(R) 64, Version `2021.10.1`. Oracle C++ Call Interface (OCCI) applications can be built with `g++8` or the `2021.10.1` Intel C++ Compiler used with the standard template libraries of `gcc8`.
Oracle XML Developer's Kit (XDK)	Oracle XML Developer's Kit is supported with the same compilers as OCCI.
Pro*COBOL	• Micro Focus Visual COBOL for Eclipse 2.3 - Update 2 • Micro Focus Visual COBOL v6.0
Unzip and Tar	Install the latest versions of Unzip and Tar. Unzip and Tar utilities are required to extract the image files for Oracle Database and Oracle Grid Infrastructure installations and patches.

Installation Requirements for Programming Environments for Linux for Arm (aarch64)

Ensure that your system meets the requirements for the programming environment you want to configure:

Table 3-10   Requirements for Programming Environments for Linux for Arm (aarch64)

Programming Environments	Support Requirements
Java Database Connectivity (JDBC) / JDBC Oracle Call Interface (JDBC OCI)	JDK 11 (Java SE Development Kit) with the JNDI extension with Oracle Java Database Connectivity.
Oracle Call Interface (OCI) Oracle C++ Call Interface Pro*C/C++	The GNU C and C++ compilers listed in this table. Oracle C++ Call Interface (OCCI) applications can be built with g++ used with the standard template libraries of the `gcc` compilers listed in this table.
`gcc` compiler packages	Install the latest released versions of the `gcc` packages listed here. gcc gcc-c++
Oracle XML Developer's Kit (XDK)	Oracle XML Developer's Kit is supported with the same compilers as OCCI.

Installation Requirements for Web Browsers

Web browsers are required only if you intend to use Oracle Enterprise Manager Cloud Control. Web browsers must support JavaScript, and the HTML 4.0 and CSS 1.0 standards.

https://support.oracle.com

Related Topics

• *Oracle Enterprise Manager Cloud Control Basic Installation Guide*

Checking Kernel and Package Requirements for Linux

Verify your kernel and packages to see if they meet minimum requirements for installation.

1. To determine the distribution and version of Linux installed, enter one of the following commands:

```
# cat /etc/oracle-release
# cat /etc/redhat-release
# cat /etc/os-release
# lsb_release -id
```

2. To determine if the required kernel errata is installed, enter the following command:

```
# uname -r
```

The following is an example of the output this command displays on an Oracle Linux 8 system:

```
5.4.17-2011.1.2.el8uek.x86_64
```

Review the required errata level for your distribution. If the errata level is previous to the required minimum errata update, then obtain and install the latest kernel update from your Linux distributor.

3. To determine whether the required packages are installed, enter commands similar to the following:

```
# rpm -q package_name
```

Alternatively, if you require specific system architecture information, then enter the following command:

```
# rpm -qa --queryformat "%{NAME}-%{VERSION}-%{RELEASE} (%{ARCH})\n" | grep package_name
```

You can also combine a query for multiple packages, and review the output for the correct versions. For example:

```
# rpm -q binutils compat-libstdc++ gcc glibc libaio libgcc libstdc++ \
make sysstat unixodbc
```

If a package is not installed, then install it from your Linux distribution media or download the required package version from your Linux distributor's website.

Verifying Digital Signature and Integrity of Installation Archive Files

Starting with Oracle Database 23ai, Oracle digitally signs the installation archive files with Oracle certificates to ensure the integrity of the packages before you deploy them in your environments.

Use the Java utility `jarsigner` to verify the integrity of your Oracle Database, Oracle Database Client, Oracle Grid Infrastructure, Oracle Examples, Oracle Gateways, or Oracle GSM installation archive files. You can verify the integrity of the installation archive files before you extract the installation files.

Quick Verification

To quickly verify the installation archive files, use the `jarsigner` command with the `-verify` option:

1. Go to the directory where you have downloaded the installation archive files.

2. Run this command to check your installation archive file:

```
jarsigner -verify installation_archive_file
```

For example, to check the Oracle Database gold image:

```
jarsigner -verify db_home.zip
```

```
jar verified.
```

Detailed Certificate Information

If you want detailed certificate information, then use the -verbose:summary and -certs along with the -verify option.

1. Go to the directory where you have downloaded the installation archive files.

2. Run this command to check your installation archive file:

```
jarsigner -verify -verbose:summary -certs installation_archive_file
```

For example, to check the Oracle Database gold image:

```
jarsigner -verify -verbose:summary -certs db_home.zip
```

The output is similar to the following:

```
2237119 Fri Feb 17 07:02:30 UTC 2023 META-INF/MANIFEST.MF

      >>> Signer
      X.509, CN="Oracle America, Inc.", O="Oracle America, Inc.",
L=Redwood City, ST=California, C=US
      [
      Signature algorithm: SHA256withRSA, 3072-bit key
      [certificate is valid from 8/19/21 12:00 AM to 8/19/23 11:59 PM]
      X.509, CN=DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1,
O="DigiCert, Inc.", C=US
      [
      Signature algorithm: SHA384withRSA, 4096-bit key
      [certificate is valid from 4/29/21 12:00 AM to 4/28/36 11:59 PM]
      X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
      [
      Signature algorithm: SHA384withRSA, 4096-bit key
      [trusted certificate]
      >>> TSA
      X.509, CN=DigiCert Timestamp 2022 - 2, O=DigiCert, C=US
      [
      Signature algorithm: SHA256withRSA, 4096-bit key
      [certificate is valid from 9/21/22 12:00 AM to 11/21/33 11:59 PM]
      X.509, CN=DigiCert Trusted G4 RSA4096 SHA256 TimeStamping CA,
O="DigiCert, Inc.", C=US
      [
      Signature algorithm: SHA256withRSA, 4096-bit key
      [certificate is valid from 3/23/22 12:00 AM to 3/22/37 11:59 PM]
      X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
      [
      Signature algorithm: SHA384withRSA, 4096-bit key
      [certificate is valid from 8/1/22 12:00 AM to 11/9/31 11:59 PM]

       2237281 Fri Feb 17 07:02:32 UTC 2023 META-INF/ORACLE_C.SF (and 1
more)

      (Signature related entries)

            0 Fri Feb 17 05:41:24 UTC 2023 OPatch/ (and 1897 more)
```

ORACLE

```
      (Directory entries)

       2977 Tue Dec 20 08:02:16 UTC 2022 OPatch/README.txt (and 20199 more)

      [entry was signed on 2/17/23 7:02 AM]
      >>> Signer
      X.509, CN="Oracle America, Inc.", O="Oracle America, Inc.",
L=Redwood City, ST=California, C=US
        [
      Signature algorithm: SHA256withRSA, 3072-bit key
      [certificate is valid from 8/19/21 12:00 AM to 8/19/23 11:59 PM]
      X.509, CN=DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1,
O="DigiCert, Inc.", C=US
        [
      Signature algorithm: SHA384withRSA, 4096-bit key
      [certificate is valid from 4/29/21 12:00 AM to 4/28/36 11:59 PM]
      X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
        [
      Signature algorithm: SHA384withRSA, 4096-bit key
      [trusted certificate]
      >>> TSA
      X.509, CN=DigiCert Timestamp 2022 - 2, O=DigiCert, C=US
        [
      Signature algorithm: SHA256withRSA, 4096-bit key
      [certificate is valid from 9/21/22 12:00 AM to 11/21/33 11:59 PM]
      X.509, CN=DigiCert Trusted G4 RSA4096 SHA256 TimeStamping CA,
O="DigiCert, Inc.", C=US
        [
      Signature algorithm: SHA256withRSA, 4096-bit key
      [certificate is valid from 3/23/22 12:00 AM to 3/22/37 11:59 PM]
      X.509, CN=DigiCert Trusted Root G4, O=DigiCert Inc, C=US
        [
      Signature algorithm: SHA384withRSA, 4096-bit key
      [certificate is valid from 8/1/22 12:00 AM to 11/9/31 11:59 PM]


  s = signature was verified
  m = entry is listed in manifest
  k = at least one certificate was found in keystore
  i = at least one certificate was found in identity scope

- Signed by "CN="Oracle America, Inc.", O="Oracle America, Inc.",
L=Redwood City, ST=California, C=US"
    Digest algorithm: SHA-256
    Signature algorithm: SHA256withRSA, 3072-bit key
  Timestamped by "CN=DigiCert Timestamp 2022 - 2, O=DigiCert, C=US" on Fri
Feb 17 07:02:33 UTC 2023
    Timestamp digest algorithm: SHA-256
    Timestamp signature algorithm: SHA256withRSA, 4096-bit key

jar verified.

The signer certificate will expire on 2023-08-19.
The timestamp will expire on 2031-11-09.
```

For more `jarsigner` options, type `jarsigner -h` or review the `jarsigner` documentation:

jarsigner Reference

4

Configuring Users, Groups and Environments for Oracle Database Client

Before installation, create operating system groups and users, and configure user environments.

- **Required Operating System Groups and Users**
  Oracle software installations require an installation owner, an Oracle Inventory group, which is the primary group of all Oracle installation owners, and at least one group designated as a system privileges group.

- **Creating Operating System Oracle Installation User Accounts**
  Before starting installation, create Oracle software owner user accounts, and configure their environments.

- **Unsetting Oracle Installation Owner Environment Variables**
  Unset Oracle installation owner environment variables before you start the installation.

Required Operating System Groups and Users

Oracle software installations require an installation owner, an Oracle Inventory group, which is the primary group of all Oracle installation owners, and at least one group designated as a system privileges group.

Review group and user options with your system administrator. If you have system administration privileges, then review the topics in this section and configure operating system groups and users as needed.

- **Determining If an Oracle Inventory and Oracle Inventory Group Exist**
  Determine if you have an existing Oracle central inventory, and ensure that you use the same Oracle Inventory for all Oracle software installations. Also, ensure that all Oracle software users you intend to use for installation have permissions to write to this directory.

- **Creating the Oracle Inventory Group If an Oracle Inventory Does Not Exist**
  Create an Oracle Inventory group manually as part of a planned installation, particularly where more than one Oracle software product is installed on servers.

- **About Oracle Installation Owner Accounts**
  Select or create an Oracle installation owner for your installation, depending on the group and user management plan you want to use for your installations.

- **Identifying an Oracle Software Owner User Account**
  You must create at least one software owner user account the first time you install Oracle software on the system. Either use an existing Oracle software user account, or create an Oracle software owner user account for your installation.

Determining If an Oracle Inventory and Oracle Inventory Group Exist

Determine if you have an existing Oracle central inventory, and ensure that you use the same Oracle Inventory for all Oracle software installations. Also, ensure that all Oracle software users you intend to use for installation have permissions to write to this directory.

When you install Oracle software on the system for the first time, OUI creates the `oraInst.loc` file. This file identifies the name of the Oracle Inventory group (by default, `oinstall`), and the path of the Oracle central inventory directory. If you have an existing Oracle central inventory, then ensure that you use the same Oracle Inventory for all Oracle software installations, and ensure that all Oracle software users you intend to use for installation have permissions to write to this directory.

```
oraInst.loccentral_inventory_locationgroup


inventory_loc=central_inventory_location
inst_group=group
```

Use the `more` command to determine if you have an Oracle central inventory on your system. For example:

```
# more /etc/oraInst.loc


inventory_loc=/u01/app/oraInventory
inst_group=oinstall
```

Use the command `grep groupname /etc/group` to confirm that the group specified as the Oracle Inventory group still exists on the system. For example:

```
$ grep oinstall /etc/group
oinstall:x:54321:grid,oracle
```

> **✎ Note:**
>
> Do not put the `oraInventory` directory under the Oracle base directory for a new installation, because that can result in user permission errors for other installations.

Creating the Oracle Inventory Group If an Oracle Inventory Does Not Exist

Create an Oracle Inventory group manually as part of a planned installation, particularly where more than one Oracle software product is installed on servers.

By default, if an oraInventory group does not exist, then the installer uses the primary group of the installation owner for the Oracle software being installed as the oraInventory group. Ensure that this group is available as a primary group for all planned Oracle software installation owners.

```
oraInst.loc


# /usr/sbin/groupadd -g 54321 oinstall
```

About Oracle Installation Owner Accounts

Select or create an Oracle installation owner for your installation, depending on the group and user management plan you want to use for your installations.

You must create a software owner for your installation in the following circumstances:

- If an Oracle software owner user does not exist; for example, if this is the first installation of Oracle software on the system.

- If an Oracle software owner user exists, but you want to use a different operating system user, with different group membership, to separate Oracle Grid Infrastructure administrative privileges from Oracle Database administrative privileges.

In Oracle documentation, a user created to own only Oracle Grid Infrastructure software installations is called the Grid user (`grid`). This user owns both the Oracle Clusterware and Oracle Automatic Storage Management binaries. A user created to own either all Oracle installations, or one or more Oracle database installations, is called the Oracle user (`oracle`). You can have only one Oracle Grid Infrastructure installation owner, but you can have different Oracle users to own different installations.

Oracle software owners must have the Oracle Inventory group as their primary group, so that each Oracle software installation owner can write to the central inventory (oraInventory), and so that OCR and Oracle Clusterware resource permissions are set correctly. The database software owner must also have the OSDBA group and (if you create them) the OSOPER, OSBACKUPDBA, OSDGDBA, OSRACDBA, and OSKMDBA groups as secondary groups.

Identifying an Oracle Software Owner User Account

You must create at least one software owner user account the first time you install Oracle software on the system. Either use an existing Oracle software user account, or create an Oracle software owner user account for your installation.

To use an existing user account, obtain from you system administrator the name of an existing Oracle installation owner. Confirm that the existing owner is a member of the Oracle Inventory group.

```
oinstalloinstall

$ grep "oinstall" /etc/group
oinstall:x:54321:oracle
```

You can then use the ID command to verify that the Oracle installation owners you intend to use have the Oracle Inventory group as their primary group. For example:`$ id oracle`

```
uid=54321(oracle) gid=54321(oinstall) groups=54321(oper),54322(dba)
```

After you create operating system groups, create or modify Oracle user accounts in accordance with your operating system authentication planning.

Creating Operating System Oracle Installation User Accounts

Before starting installation, create Oracle software owner user accounts, and configure their environments.

Oracle software owner user accounts require resource settings and other environment configuration. To protect against accidents, Oracle recommends that you create one software installation owner account for each Oracle software program you install.

- Creating an Oracle Software Owner User
  If the Oracle software owner user (`oracle`) does not exist, or if you require a new Oracle software owner user, then create it as described in this section.

- Environment Requirements for Oracle Software Owners
  You must make the following changes to configure Oracle software owner environments:

- Procedure for Configuring Oracle Software Owner Environments
  Configure each Oracle installation owner user account environment:

- Setting Remote Display and X11 Forwarding Configuration
  If you are on a remote terminal, and the local system has only one visual (which is typical), then use the following syntax to set your user account `DISPLAY` environment variable:

Creating an Oracle Software Owner User

If the Oracle software owner user (`oracle`) does not exist, or if you require a new Oracle software owner user, then create it as described in this section.

The following example shows how to create the user `oracle` with the user ID 54321; with the primary group `oinstall`; and with secondary group `dba`.

```
# /usr/sbin/useradd -u 54321 -g oinstall -G dba oracle
```

You must note the user ID number for installation users, because you need it during preinstallation.

For Oracle Grid Infrastructure Installations, user IDs and group IDs must be identical on all candidate nodes.

Environment Requirements for Oracle Software Owners

You must make the following changes to configure Oracle software owner environments:

- Set the installation software owner user (`grid`, `oracle`) default file mode creation mask (umask) to 022 in the shell startup file. Setting the mask to 022 ensures that the user performing the software installation creates files with 644 permissions.

- Set ulimit settings for file descriptors and processes for the installation software owner (`grid`, `oracle`).

- Set the DISPLAY environment variable in preparation for running an Oracle Universal Installer (OUI) installation.

> ⚠️ **Caution:**
>
> If you have existing Oracle installations that you installed with the user ID that is your Oracle Grid Infrastructure software owner, then unset all Oracle environment variable settings for that user.

Procedure for Configuring Oracle Software Owner Environments

Configure each Oracle installation owner user account environment:

1. Start an X terminal session (`xterm`) on the server where you are running the installation.

2. Enter the following command to ensure that X Window applications can display on this system, where hostname is the fully qualified name of the local host from which you are accessing the server:

   ```
   $ xhost + hostname
   ```

3. If you are not logged in as the software owner user, then switch to the software owner user you are configuring. For example, with the user `grid`:

   ```
   $ su - grid
   ```

   On systems where you cannot run `su` commands, use `sudo` instead:

   ```
   $ sudo -u grid -s
   ```

4. To determine the default shell for the user, enter the following command:

   ```
   $ echo $SHELL
   ```

5. Open the user's shell startup file in any text editor:

   - Bash shell (`bash`):

     ```
     $ vi .bash_profile
     ```

   - Bourne shell (`sh`) or Korn shell (`ksh`):

     ```
     $ vi .profile
     ```

   - C shell (`csh` or `tcsh`):

     ```
     % vi .login
     ```

6. Enter or edit the following line, specifying a value of `022` for the default file mode creation mask:

   ```
   umask 022
   ```

7. If the `ORACLE_SID`, `ORACLE_HOME`, or `ORACLE_BASE` environment variables are set in the file, then remove these lines from the file.

8. Save the file, and exit from the text editor.

9. To run the shell startup script, enter one of the following commands:

   - Bash shell:

     ```
     $ . ./.bash_profile
     ```

- Bourne, Bash, or Korn shell:

  ```
  $ . ./.profile
  ```

- C shell:

  ```
  % source ./.login
  ```

10. Use the following command to check the PATH environment variable:

    ```
    $ echo $PATH
    ```

    Remove any Oracle environment variables.

11. Unset any Oracle environment variables.

    If you have an existing Oracle software installation, and you are using the same user to install this installation, then unset the $ORACLE_HOME, $ORA_NLS10, and $TNS_ADMIN environment variables.

    If you have set $ORA_CRS_HOME as an environment variable, then unset it before starting an installation or upgrade. Do not use $ORA_CRS_HOME as a user environment variable, except as directed by Oracle Support.

12. If you are not installing the software on the local system, then enter a command similar to the following to direct X applications to display on the local system:

    - Bourne, Bash, or Korn shell:

      ```
      $ export DISPLAY=local_host:0.0
      ```

    - C shell:

      ```
      % setenv DISPLAY local_host:0.0
      ```

    In this example, `local_host` is the host name or IP address of the system (your workstation, or another client) on which you want to display the installer.

13. If the `/tmp` directory has less than 1 GB of free space, then identify a file system with at least 1 GB of free space and set the `TMP` and `TMPDIR` environment variables to specify a temporary directory on this file system:

    > **Note:**
    >
    > You cannot use a shared file system as the location of the temporary file directory (typically `/tmp`) for Oracle RAC installations. If you place `/tmp` on a shared file system, then the installation fails.

    a. Use the `df -h` command to identify a suitable file system with sufficient free space.

b. If necessary, enter commands similar to the following to create a temporary directory on the file system that you identified, and set the appropriate permissions on the directory:

```
$ sudo -s
# mkdir /mount_point/tmp
# chmod 775 /mount_point/tmp
# exit
```

c. Enter commands similar to the following to set the `TMP` and `TMPDIR` environment variables:

Bourne, Bash, or Korn shell:

```
$ TMP=/mount_point/tmp
$ TMPDIR=/mount_point/tmp
$ export TMP TMPDIR
```

C shell:

```
% setenv TMP /mount_point/tmp
% setenv TMPDIR /mount_point/tmp
```

14. To verify that the environment has been set correctly, enter the following commands:

```
$ umask
$ env | more
```

Verify that the `umask` command displays a value of `22`, `022`, or `0022` and that the environment variables you set in this section have the correct values.

Setting Remote Display and X11 Forwarding Configuration

If you are on a remote terminal, and the local system has only one visual (which is typical), then use the following syntax to set your user account `DISPLAY` environment variable:

Remote Display

```
Bourne, Korn, and Bash shells
```

```
$ export DISPLAY=hostname:0
```

```
C shell
```

```
% setenv DISPLAY hostname:0
```

For example, if you are using the Bash shell and if your host name is `local_host`, then enter the following command:

```
$ export DISPLAY=node1:0
```

X11 Forwarding

To ensure that X11 forwarding does not cause the installation to fail, use the following procedure to create a user-level SSH client configuration file for Oracle installation owner user accounts:

1. Using any text editor, edit or create the software installation owner's `~/.ssh/config file`.

2. Ensure that the `ForwardX11` attribute in the `~/.ssh/config` file is set to `no`. For example:

```
Host *
    ForwardX11 no
```

3. Ensure that the permissions on `~/.ssh` are secured to the Oracle installation owner user account. For example:

```
$ ls -al .ssh
total 28
drwx------  2 grid oinstall 4096 Jun 21 2020
drwx------ 19 grid oinstall 4096 Jun 21 2020
-rw-r--r--  1 grid oinstall 1202 Jun 21 2020 authorized_keys
-rwx------  1 grid oinstall  668 Jun 21 2020 id_dsa
-rwx------  1 grid oinstall  601 Jun 21 2020 id_dsa.pub
-rwx------  1 grid oinstall 1610 Jun 21 2020 known_hosts
```

Unsetting Oracle Installation Owner Environment Variables

Unset Oracle installation owner environment variables before you start the installation.

The environment variables you have set for the Oracle installation owner account you use to run the installation can cause issues if they are set to values that conflict with the values needed for installation.

If you have set ORA_CRS_HOME as an environment variable, following instructions from Oracle Support, then unset it before starting an installation or upgrade. You should never use ORA_CRS_HOME as an environment variable except under explicit direction from Oracle Support.

If you have had an existing installation on your system, and you are using the same user account to install this installation, then unset the following environment variables: ORA_CRS_HOME, ORACLE_HOME, ORA_NLS10, TNS_ADMIN, and any other environment variable set for the Oracle installation user that is connected with Oracle software homes.

Also, ensure that the `$ORACLE_HOME/bin` path is removed from your PATH environment variable.

5
Installing Oracle Database Client

The Oracle Database Client software is available as an image file for download and installation. Oracle Database Client installation binaries continue to be available in the traditional format as non-image zip files.

You can download Oracle Database software from the Oracle website or the Oracle Software Delivery Cloud portal. In most cases, you use the graphical user interface (GUI) provided by the installer to install the software. However, you can also run silent mode installations, without using the GUI.

> **Note:**
>
> You cannot use the installer from an earlier Oracle release to install components from this release.

- About Image-Based Oracle Database Client Installation
  Starting with Oracle Database 19c, installation and configuration of Oracle Database Client software is simplified with image-based installation.

- Downloading Oracle Software
  Select the method you want to use to download the software.

- About Character Set Selection During Installation
  Before you create the database, decide the character set that you want to use.

- Running the Installer in a Different Language
  Describes how to run the installer in other languages.

- Installing the Oracle Database Client Software
  These topics explain how to run the Setup Wizard to perform most database client installations.

- Relinking Oracle Database Client Binaries After Installation
  After an Oracle Database Client installation, if required, you can modify the binaries using the `relink as_installed` option.

About Image-Based Oracle Database Client Installation

Starting with Oracle Database 19c, installation and configuration of Oracle Database Client software is simplified with image-based installation.

To install Oracle Database Client, create the new Oracle home, extract the image file into the newly-created Oracle home, and run the setup wizard to register the Oracle Database product.

You must extract the image software (`client_home.zip`) into the directory where you want your Oracle Database Client home to be located, and then run the Setup Wizard to start the Oracle Database Client installation and configuration. Oracle recommends that the Oracle home directory path you create is in compliance with the Oracle Optimal Flexible Architecture recommendations.

As with Oracle Database and Oracle Grid Infrastructure image file installations, Oracle Database Client image installations simplify Oracle Database Client installations and ensure best practice deployments. Oracle Database Client installation binaries continue to be available in the traditional format as non-image zip files.

Using image-based installation, you can install Oracle Database Client 64-bit configurations of the Administrator installation type.

32-bit Oracle Database clients are desupported in Oracle Database 23ai.

Oracle has discontinued developing 32-bit Oracle Database clients. Oracle recommends that you use 64-bit Oracle Database clients. If you are using 32-bit applications, then you can continue to use older 32-bit Oracle Database clients, subject to "Client / Server Interoperability Support Matrix for Different Oracle Versions (Doc ID 207303.1)," and the release support lifecycle (My Oracle Support 742060.1).

Downloading Oracle Software

Select the method you want to use to download the software.

You can download Oracle Database software from the Oracle website or the Oracle Software Delivery Cloud portal and extract them to the Oracle home. Ensure that you review and understand the terms of the license.

- Downloading the Installation Archive Files from the Oracle Database Website
  Download the installation archive files from the Oracle Database downloads page.

- Downloading the Software from Oracle Software Delivery Cloud Portal
  You can download the software from Oracle Software Delivery Cloud.

Downloading the Installation Archive Files from the Oracle Database Website

Download the installation archive files from the Oracle Database downloads page.

1. Use any browser to access the Oracle Database software downloads page:

   https://www.oracle.com/database/technologies/oracle-database-software-downloads.html

2. Go to the download page for the product to install.

3. On the download page, identify the required disk space by adding the file sizes for each required file.

   The file sizes are listed next to the file names.

4. Select a file system with enough free space to store and expand the archive files.

   In most cases, the available disk space must be at least twice the size of all of the archive files.

5. On the file system, create a parent directory for each product (for example, `OraDB23ai`) to hold the installation directories.

6. Download all of the installation archive files to the directory you created for the product.

> **✎ Note:**
>
> For Oracle Database Client installations, there are two installation archive files available for download. The first file is the client installation binary and the second file is a client gold image file. Download the appropriate zip file based on the type of installation you want to perform.

7. Verify that the files you downloaded are the same size as the corresponding files on the Oracle website. Also verify the checksums are the same as noted on the Oracle website using a command similar to the following, where *filename* is the name of the file you downloaded:

```
cksum filename.zip
```

8. Extract the files in each directory that you just created.

Downloading the Software from Oracle Software Delivery Cloud Portal

You can download the software from Oracle Software Delivery Cloud.

1. Use a browser to access the Oracle Software Delivery Cloud portal:

   https://edelivery.oracle.com/

2. Click **Sign In** and enter your Oracle account username and password.

3. Type **Oracle Database** in the search bar. Click the **Add to Cart** button corresponding to the Oracle Database version that you want to download

4. In the Checkout page, click **Checkout** and deselect any products that you do not want to download.

5. Select the operating system platform on which you want to install the software from the **Platform/Languages** column.

6. Click **Continue**.

7. Review the license agreement.

8. Select the **I reviewed and accept the Oracle License Agreement** checkbox. Click **Continue**.

9. Click **Download** to start downloading the software.

10. After you download the files, click **View Digest** to verify that the checksum matches the value listed on the download page.

About Character Set Selection During Installation

Before you create the database, decide the character set that you want to use.

After a database is created, changing its character set is usually very expensive in terms of time and resources. Such operations may require converting all character data by exporting the whole database and importing it back. Therefore, it is important that you carefully select the database character set at installation time.

Oracle Database uses character sets for the following:

- Data stored in SQL character data types (`CHAR`, `VARCHAR2`, `CLOB`, and `LONG`).

- Identifiers such as table names, column names, and PL/SQL variables.

- Stored SQL and PL/SQL source code, including text literals embedded in this code.

The default database character set of a database created from the General Purpose/ Transaction Processing or the Data Warehousing template is Unicode `AL32UTF8`.

Unicode is the universal character set that supports most of the currently spoken languages of the world. It also supports many historical scripts (alphabets). Unicode is the native encoding of many technologies, including Java, XML, XHTML, ECMAScript, and LDAP. Unicode is ideally suited for databases supporting the Internet and the global economy.

Because `AL32UTF8` is a multibyte character set, it requires slightly more CPU time for text processing compared to single-byte character sets. Also, storage space requirements are higher for text in most languages compared to corresponding legacy character sets. However, the universality and flexibility of Unicode that enables addition of data in new languages to applications running in an `AL32UTF8` database generally outweighs these additional costs.

The database character set of an Oracle Database, that is, of its `CDB$ROOT` container, determines which pluggable databases (PDBs) can be plugged into it. If you use Unicode `AL32UTF8` as your database character set, then you can plug in a PDB in any database character set supported by Oracle Database (with the exception of `EBCDIC`-based character sets). If you use any character set other than `AL32UTF8` when creating the container database, you will be able to plug in PDBs in the same character set only. Therefore, you should generally use the default option for the database character set when installing a new database.

If you need to deploy PDBs in a given legacy character set to fulfill a specific compatibility, storage, or performance requirement, create a temporary container database in this legacy character set with one empty PDB. This PDB will have the same legacy database character set. Then, unplug this PDB and plug it into the target `AL32UTF8` container database. Drop the temporary container database. You can use such a plugged-in PDB as a template to clone further PDBs in the same legacy character set as needed. You can use the same method to add further legacy character set template PDBs to the same `AL32UTF8` container database, as required.

> ✎ **See Also:**
>
> *Oracle Database Globalization Support Guide* for more information about choosing a database character set for an Oracle Database.

Running the Installer in a Different Language

Describes how to run the installer in other languages.

Your operating system locale determines the language in which the database installer runs. You can run the installer in one of these languages:

- Brazilian Portuguese (pt_BR)

- French (fr)

- German (de)

- Italian (it)

- Japanese (ja)

- Korean (ko)

- Simplified Chinese (zh_CN)

- Spanish (es)

- Traditional Chinese (zh_TW)

To run the database installer in a supported language, change the locale in which your operating system session is running before you start the installer.

If the selected language is not one of the supported languages, then the installer runs in English.

Installing the Oracle Database Client Software

These topics explain how to run the Setup Wizard to perform most database client installations.

The Oracle Database Client software is also available as an image file for download and installation. Oracle Database client installation binaries continue to be available in the traditional format as non-image zip files.

Starting with Oracle Database Client 23ai, the Setup Wizard no longer provides the option to install Oracle Instant Client.

- Running Setup Wizard to Install Oracle Database Client
  Use the `runInstaller` command to start the Oracle Database Client installation.

- Installing Oracle Database Client Using an Image File
  Extract the Oracle Database Client image files and use the `runInstaller` command to start the Oracle Database Client installation.

- Installing Oracle Connection Manager Using an Image File
  Starting with Oracle Database Client 23ai, Oracle Connection Manager is available as an image file for installation and configuration.

- Moving an Existing Oracle Connection Manager to the Latest Version
  Learn how to move an existing Oracle Connection Manager (CMAN) 23ai version to the latest Oracle Connection Manager version.

- Using Oracle Net Configuration Assistant
  Run Oracle Net Configuration Assistant in standalone mode after the Oracle Database Client installation is complete to configure the listener, naming methods, net service names, and directory server usage.

Running Setup Wizard to Install Oracle Database Client

Use the `runInstaller` command to start the Oracle Database Client installation.

Have all the information you need to provide regarding users groups, and storage paths before you start the installation.

During installation, you are asked to run configuration scripts as the `root` user. You can either run these scripts manually as `root` when prompted, or you can provide configuration information and passwords using a root privilege delegation option such as `sudo`.

1. Log in as the Oracle installation owner user account (`oracle`).

2. From where you have downloaded the installation binaries, run the `runInstaller` command to start the Oracle setup wizard.

For example:

```
$ cd /home/oracle_sw/
$ ./runInstaller
```

> **Note:**
>
> • Run the `runInstaller` command from the Oracle home directory only. Do not use the `runInstaller` command that resides at `$ORACLE_HOME/oui/bin/`, or any other location, to install Oracle Database, Oracle Database Client, or Oracle Grid Infrastructure.
>
> • Oracle home or Oracle base cannot be `symlinks`, nor can any of their parent directories, all the way to up to the `root` directory.

3. Select your installation type.

   Installation screens vary depending on the installation option you select. Respond to the configuration prompts as needed.

   > **Note:**
   >
   > At any time during installation, if you have a question about what you are being asked to do, click **Help**.

Installing Oracle Database Client Using an Image File

Extract the Oracle Database Client image files and use the `runInstaller` command to start the Oracle Database Client installation.

Starting with 19c, the Oracle Database Client software is also available as an image file for download and installation.

Have all the information you need to provide regarding storage paths before you start the installation. Oracle recommends that you have your My Oracle Support credentials available during installation. During installation, you are asked to run configuration scripts as the `root` user. You must run these scripts manually as `root` when prompted.

1. Log in as the Oracle installation owner user account (`oracle`).

2. Download the Oracle Database Client installation image files (`client_home.zip`) to a directory of your choice. For example, you can download the image files to the `/tmp` directory.

3. Create the Oracle home directory and extract the image files that you have downloaded in to this Oracle home directory. For example:

```
$ mkdir -p /u01/app/oracle/product/23.0.0/client_1
$ chgrp oinstall /u01/app/oracle/product/23.0.0/client_1
$ cd /u01/app/oracle/product/23.0.0/client_1
$ unzip -q /tmp/client_home.zip
```

> **Note:**
>
> - Oracle recommends that the Oracle home directory path you create is in compliance with the Oracle Optimal Flexible Architecture recommendations. Also, unzip the installation image files only in this Oracle home directory that you created.
>
> - Oracle home or Oracle base cannot be `symlinks`, nor can any of their parent directories, all the way to up to the `root` directory.

4. From the Oracle home directory, run the `runInstaller` command to start the Oracle Database Client Setup Wizard.

```
$ cd /u01/app/oracle/product/23.0.0/client_1
$ ./runInstaller
```

> **Note:**
>
> Run the `runInstaller` command from the Oracle home directory only. Do not use the `runInstaller` command that resides at `$ORACLE_HOME/oui/bin/`, or any other location, to install Oracle Database, Oracle Database Client, or Oracle Grid Infrastructure.

5. The setup wizard starts an Administrator type installation of Oracle Database Client.

   Installation screens vary depending on the installation option you select. Respond to the configuration prompts as needed.

> **Note:**
>
> At any time during installation, if you have a question about what you are being asked to do, click **Help**.

Installing Oracle Connection Manager Using an Image File

Starting with Oracle Database Client 23ai, Oracle Connection Manager is available as an image file for installation and configuration.

You must extract the image software (`client_cman_home.zip`) into the directory where you want your Oracle Database Client home to be located, and then run the Setup Wizard to start the Oracle Connection Manager installation and configuration.

Have all the information you need to provide regarding storage paths before you start the installation. During installation, you are asked to run configuration scripts as the `root` user. You must run these scripts manually as `root` when prompted.

1. Log in as the Oracle installation owner user account (`oracle`).

2. Download the Oracle Database Client installation image files (`client_cman_home.zip`) to a directory of your choice. For example, you can download the image files to the `/tmp` directory.

3. Create the Oracle home directory and extract the image files that you have downloaded in to this Oracle home directory. For example:

```
mkdir -p /u01/app/oracle/product/23.0.0/client_1
chgrp oinstall /u01/app/oracle/product/23.0.0/client_1
cd /u01/app/oracle/product/23.0.0/client_1
unzip -q /tmp/client_cman_home.zip
```

> **Note:**
>
> • Oracle recommends that the Oracle home directory path you create is in compliance with the Oracle Optimal Flexible Architecture recommendations. Also, unzip the installation image files only in this Oracle home directory that you created.
>
> • Oracle home or Oracle base cannot be `symlinks`, nor can any of their parent directories, all the way to up to the `root` directory.

4. From the Oracle home directory, run the `runInstaller` command to start the Oracle Database Client Setup Wizard.

```
$ cd /u01/app/oracle/product/23.0.0/client_1
$ ./runInstaller
```

> **Note:**
>
> Run the `runInstaller` command from the Oracle home directory only. Do not use the `runInstaller` command that resides at `$ORACLE_HOME/oui/bin/`, or any other location, to install Oracle Database, Oracle Database Client, or Oracle Grid Infrastructure.

5. The setup wizard starts the Oracle Connection Manager type installation of Oracle Database Client.

Respond to the configuration prompts as needed to provide information such as the Oracle base and Oracle inventory.

> **Note:**
>
> At any time during installation, if you have a question about what you are being asked to do, click **Help**.

Moving an Existing Oracle Connection Manager to the Latest Version

Learn how to move an existing Oracle Connection Manager (CMAN) 23ai version to the latest Oracle Connection Manager version.

1. Install the latest version of Oracle Connection Manager 23ai using an image file in the new Oracle home. See Installing Oracle Connection Manager Using an Image File.

2. Stop the existing `cman` service using the Oracle Connection Manager Control utility (`cmctl`). For example, for the 23.5 Oracle home, at the command line, enter `cmctl` with no arguments to obtain the `CMCTL` prompt, and then run the `ADMINISTER` and `SHUTDOWN` commands.

```
./cmctl
CMCTL> ADMINISTER [cman_instance]
CMCTL> SHUTDOWN
```

3. Move the existing configuration file (`cman.ora`) to the new Oracle home.

4. Start the `cman` services from the new Oracle home using the Oracle Connection Manager Control utility (`cmctl`). For example, from the 23.x Oracle home, at the command line, enter `cmctl` with no arguments to obtain the `CMCTL` prompt, and then run the `ADMINISTER` and `STARTUP` commands.

```
./cmctl
CMCTL> ADMINISTER [cman_instance]
CMCTL> STARTUP
```

Using Oracle Net Configuration Assistant

Run Oracle Net Configuration Assistant in standalone mode after the Oracle Database Client installation is complete to configure the listener, naming methods, net service names, and directory server usage.

Oracle recommends that you have information ready about the host name of the computer where the Oracle database is installed.

To start Oracle Net Configuration Assistant in standalone mode:

1. Run `netca` from the `$ORACLE_HOME/bin` directory.

2. Respond to the configuration prompts and screens as needed. The screens vary depending on the options you select. At any time during the configuration, if you have a question about what you are being asked to do, click **Help.**

Related Topics

• *Oracle Database Net Services Administrator's Guide*

Relinking Oracle Database Client Binaries After Installation

After an Oracle Database Client installation, if required, you can modify the binaries using the `relink as_installed` option.

For example, you might want to relink the Oracle Database Client binaries every time you apply an operating system patch or after an operating system upgrade.

> ⚠️ **Caution:**
>
> Before relinking executables, you must shut down all executables that run in the Oracle home directory that you are relinking. In addition, shut down applications linked with Oracle shared libraries.

1. Login as the Oracle Database Client owner user (`oracle`).

2. Set the ORACLE_HOME environment variable

   ```
   $ ORACLE_HOME=/u01/app/oracle/product/23.0.0/client_1
   ```

3. Go to the `$ORACLE_HOME/bin` directory:

   ```
   $ cd $ORACLE_HOME/bin
   ```

4. Run the `relink` script with the `as_installed` option to relink the binaries.

   ```
   $ ./relink as_installed
   ```

   The relinking is complete and the log files are generated under the `$ORACLE_HOME/install` directory.

6

Installing Oracle Instant Client

You can install Oracle Instant Client by downloading either the zip files or RPMs from the Oracle Instant Client download page on the Oracle website.

Oracle Instant Client is available on Linux x86-64 and Linux for Arm (aarch64) platforms.

- About Oracle Instant Client
  Learn how Oracle Instant Client enables applications to connect to a local or remote Oracle Database for development and production deployment.

- Operating System Requirements for Linux for Arm (aarch64)
  The Linux distribution and packages listed in this section are supported for this release on Linux for Arm (aarch64).

- Verifying Oracle Instant Client Code Signing
  Starting with Oracle Database Client 19c, you can verify the code sign for Oracle Instant Client zips and RPMs to ensure the integrity of the packages before you deploy them in your environments.

- Installing Oracle Instant Client Packages
  Learn the different methods to install Oracle Instant Client on Linux x86-64 and Linux for Arm (aarch64) platforms.

- Installing Oracle Instant Client Basic Light
  Learn how to install Oracle Instant Client Basic Light on Linux x86-64 and Linux for Arm (aarch64) platforms.

- Instant Client Libraries for OCI
  Learn about the Oracle Database client-side files used by Oracle Call Interface (OCI) applications.

- Environment Variables for Oracle Instant Client
  Learn about some common Oracle environment variables that can influence Oracle Instant Client installations.

- About the Oracle Instant Client SDK
  Learn about the Software Development Kit (SDK) development tools for Oracle Instant Client.

- Patching Oracle Instant Client Shared Libraries
  Learn how to patch Oracle Instant Client shared libraries with a complete Oracle Database Client installation.

About Oracle Instant Client

Learn how Oracle Instant Client enables applications to connect to a local or remote Oracle Database for development and production deployment.

The Oracle Instant Client libraries provide the necessary network connectivity, as well as Oracle Database client-side files to create and run Oracle Call Interface (OCI), Oracle C++ Call Interface (OCCI), ODBC, and JDBC OCI applications to make full use of Oracle Database.

Oracle Instant Client is commonly used by the Oracle APIs of popular languages and environments including Python, Node.js, Go, Ruby, and PHP. Tools included in Oracle Instant

Client, such as SQL*Plus, SQL*Loader, and Oracle Data Pump provide quick and convenient data access. Oracle Instant Client simplifies the deployment of applications by eliminating the need for an Oracle home on the client machines.

The Oracle JDBC Oracle Call Interface (OCI) Type 2 client driver (also known as a "thick" driver) is deprecated in Oracle Database 23ai.

Most Java applications that use Open Database Connectivity (ODBC) with Oracle JDBC drivers use the Thin driver. To enable Oracle to allocate resources to better address customer requirements, Oracle is deprecating the JDBC-OCI driver.

Oracle's standard client-server network interoperability allows connections between different versions of Oracle Instant Client and Oracle Database. For certified configurations, refer to My Oracle Support note 207303.1. Tools such as Data Pump may have other restrictions.

Oracle Instant Client supports the Federal Information Processing Standard (FIPS) for 140-2. For more information about configuring FIPS on Oracle Instant Client, see Oracle Database Security Guide.

Benefits of Oracle Instant Client

- Easy installation by extracting and installing zip files or RPM packages.

- Storage space requirement of applications running in Instant Client-mode is significantly reduced compared to the same application running in a full client-side installation.

- No loss of functionality or performance for applications deployed using Oracle Instant Client.

- Easy for independent software vendors to package applications.

Oracle Instant Client Packages

Various Oracle Instant Client packages are available that help you run applications.

Table 6-1   Oracle Instant Client Packages

Package Name	Description
Basic	All files required to run Oracle Call Interface (OCI), OCCI, and JDBC-OCI applications for Oracle Database
Basic Light	Smaller version of the Basic package, with only English error messages and Unicode, ASCII, and Western European character set support.
SDK	Additional header files and an example `makefile` for developing OCI and OCCI applications.
SQL*Plus	Additional package providing the SQL*Plus command-line tool for executing SQL and PL/SQL statements and scripts.
Tools	Additional tools including Data Pump, SQL*Loader and Workload Replay Client.
ODBC	Additional libraries providing ODBC.
Precompilers	Additional tools and libraries providing the Pro*C and Pro*COBOL precompilers.
JDBC-OCI Supplement	Additional libraries to support Internationalization.

Oracle Instant Client Utilities

Oracle Instant Client and Oracle Instant Client Basic Light contain the following utilities:

- **genezi**: You can retrieve information about your Oracle Instant Client installation using the `genezi` utility. This utility displays information such as the time zone files that are part of the Oracle Instant Client data shared library. The timezone information shows either the embedded timezone files, or those specified by the ORA_TZFILE environment variable, if it is set.

  Run the `genezi -v` command to determine the client library information.

- **adrci**: Automatic Diagnostic Repository Command Interpreter (`adrci`) is a command-line tool that you can use to manage Oracle Database diagnostic data. `adrci` enables you to investigate problems, view health check reports, and package first-failure diagnostic data, all within a command-line environment.

- **uidrvci**: Do not use the `uidrvci.exe` file as it is used to access diagnostic data.

Operating System Requirements for Linux for Arm (aarch64)

The Linux distribution and packages listed in this section are supported for this release on Linux for Arm (aarch64).

Identify the requirements for your Linux distribution, and ensure that you have a supported kernel and required packages installed before starting installation.

The platform-specific hardware and software requirements included in this guide were current when this guide was published. However, because new platforms and operating system software versions may be certified after this guide is published, review the certification matrix on the My Oracle Support website for the most up-to-date list of certified hardware platforms and operating system versions:

https://support.oracle.com/

> **Note:**
>
> Oracle Database 23ai is currently not supported on non-Oracle hardware. For support, availability, and database offerings, see the *Oracle Database Licensing Information User Manual* and My Oracle Support Note 742060.1.

- Supported Oracle Linux 8 Distributions on Linux for Arm (aarch64)
  Use the following information to check supported Oracle Linux 8 distributions for Oracle Instant Client on Linux for Arm (aarch64).

Supported Oracle Linux 8 Distributions on Linux for Arm (aarch64)

Use the following information to check supported Oracle Linux 8 distributions for Oracle Instant Client on Linux for Arm (aarch64).

Table 6-2   Linux for Arm (aarch64) Oracle Linux 8 Minimum Operating System Requirements for Oracle Instant Client

Item	Requirements
Oracle Linux 8	Starting with Oracle Database 23ai Release 23.7, the Linux for Arm (aarch64) operating system is supported on Oracle Instant Client with the following minimum supported version:
	Oracle Linux 8.6 with Unbreakable Enterprise Kernel 7: 5.15.0-205.149.5.1.el8uek.aarch64
	Note: Oracle recommends that you update Oracle Linux 8 to the latest available release level.
Packages for Oracle Linux 8	Install the latest released versions of the following packages:
	Subscribe to the Oracle Linux 8 channel on the Unbreakable Linux Network, or configure a yum repository from the Oracle Linux yum server website, and then install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`. The Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, automatically installs all required packages listed in the table below, their dependencies for Oracle Grid Infrastructure and Oracle Database installations, and also performs other system configuration. If you install the Oracle Database Preinstallation RPM, `oracle-database-preinstall-23ai`, then you do not have to install these packages, as the Oracle Database Preinstallation RPM automatically installs them.
	bc binutils elfutils-libelf gcc gcc-c++ glibc glibc-devel ksh libaio libaio-devel libgcc libgfortran libibverbs libnsl libnsl2 libstdc++ libstdc++-devel libxcb libX11 libXau libXi libXrender libXtst make policycoreutils policycoreutils-python-utils smartmontools sysstat

Verifying Oracle Instant Client Code Signing

Starting with Oracle Database Client 19c, you can verify the code sign for Oracle Instant Client zips and RPMs to ensure the integrity of the packages before you deploy them in your environments.

- Verifying Signed Oracle Instant Client Zip Files
  Use the Java utility `jarsigner` to verify the integrity of your Oracle Instant Client zip files.

- Verifying Signed Oracle Instant Client RPM Packages
  Use the `rpm` command to perform operations such as listing and verifying the signature fingerprint for Oracle Instant Client RPM packages for Oracle Linux.

Verifying Signed Oracle Instant Client Zip Files

Use the Java utility `jarsigner` to verify the integrity of your Oracle Instant Client zip files.

To verify the integrity of the Oracle Instant Client installation zip files before you extract the installation files:

1. Go to the directory where you have downloaded the zip file.

2. Run the `jarsigner` command with the `-verify` option to quickly check your installation zip file:

   ```
   jarsigner -verify instant_client_zip_file
   ```

   For example, to check the Oracle Instant Client Basic package zip file for **Linux x86-64**:

   ```
   jarsigner -verify instantclient-basic-linux.x64-23.4.0.24.05.zip
   jar verified.
   ```

3. If you want detailed certificate information, then use the `-verify` option along with the `-verbose:summary` and `-certs` options. For example:

   ```
   jarsigner -verify -verbose:summary -certs instantclient-basic-
   linux.x64-23.4.0.24.05.zip
   ```

   For more `jarsigner` options, type `jarsigner -h` or review the jarsigner reference documentation.

Verifying Signed Oracle Instant Client RPM Packages

Use the `rpm` command to perform operations such as listing and verifying the signature fingerprint for Oracle Instant Client RPM packages for Oracle Linux.

Ensure that the RPM signature fingerprint matches the expected key fingerprint and also verify the RPM package has a signature. Before you start this process, perform the following tasks:

1. List the installed keys on your system:

   ```
   rpm -qa gpg-pubkey
   ```

The output of this command is similar to the following:

```
gpg-pubkey-ad986da3-5cabf60d
gpg-pubkey-16c083cd-49af3996
```

2. Import the GPG keys (optional):

```
rpm --import GPG-KEY
```

> **Note:**
>
> Depending on your verification purpose you can install any GPG keys in the RPM database on your system to verify the signed packages.

Scenario 1: To verify an RPM package that has a signature with the key not installed on the system by default

1. Run the `rpm -qip` command to check the Oracle Instant Client Basic RPM package fingerprint. For example, on Oracle Linux 8 for **Linux x86-64**:

```
rpm -qip oracle-instantclient-basic-23.4.0.24.05-1.el8.x86_64.rpm | grep
Signature
```

The output of this command is similar to the following:

```
warning: oracle-instantclient-basic-23.4.0.24.05-1.el8.x86_64.rpm: Header
V3 RSA/SHA256 Signature, key ID 8d8b756f: NOKEY
Signature: RSA/SHA256, Mon 09 Jan 2023 07:38:06 PM GMT, Key ID
bc4d06a08d8b756f
```

2. Run the `rpm --checksig --verbose` command to verify the package and print headers statuses and fingerprint details for the Oracle Instant Client Basic RPM package. For example, for **Linux x86-64**:

```
rpm --checksig --verbose oracle-instantclient-
basic-23.4.0.24.05-1.el8.x86_64.rpm
```

The output of this command is similar to the following:

```
oracle-instantclient-basic-23.4.0.24.05-1.el8.x86_64.rpm:
Header V3 RSA/SHA256 Signature, key ID 8d8b756f: NOKEY
```

This output verifies the Oracle Instant Client Basic RPM package signature by printing the key ID and returning `NOKEY` verification status as the GPG key with this ID is not present in the RPM key management system.

Scenario 2: To verify RPM package with both signature and key installed

1. Run the `rpm -qip` command to check the Oracle Instant Client Basic RPM package fingerprint. For example, for **Linux x86-64**:

    ```
    rpm -qip oracle-instantclient-basic-23.4.0.24.05-1.el8.x86_64.rpm | grep
    Signature
    ```

    The output of this command is similar to the following:

    ```
    Signature: RSA/SHA256, Fri 06 Jan 2023 11:11:04 PM GMT, Key ID
    82562ea9ad986da3
    ```

2. Run the `rpm --checksig --verbose` command to verify the package and print headers statuses and fingerprint details for Oracle Instant Client Basic RPM package. For example, for **Linux x86-64**:

    ```
    rpm --checksig --verbose oracle-instantclient-
    basic-23.4.0.24.05-1.el8.x86_64.rpm
    ```

    The output of this command is similar to the following:

    ```
    oracle-instantclient-basic-23.4.0.24.05-1.el8.x86_64.rpm:
    Header V3 RSA/SHA256 Signature, key ID ad986da3: OK
        Header SHA256 digest: OK
        Header SHA1 digest: OK
        Payload SHA256 digest: OK
        V3 RSA/SHA256 Signature, key ID ad986da3: OK
        MD5 digest: OK
    ```

    This output verifies the Oracle Instant Client Basic RPM package signature with the key and displays the fingerprint.

Related Topics

*   Oracle Linux Keys
*   yum.oracle.com FAQs

Installing Oracle Instant Client Packages

Learn the different methods to install Oracle Instant Client on Linux x86-64 and Linux for Arm (aarch64) platforms.

> **Note:**
>
> Starting with Oracle Database Client 23ai, the Setup Wizard in the full client installation no longer provides the option to install Oracle Instant Client.

*   Installing Oracle Instant Client Using Zip Files
    Learn how to download and use zip files to install Oracle Instant Client on Linux x86-64 and Linux for Arm (aarch64) platforms.

- Installing Oracle Instant Client Using RPMs
  Learn how to install Oracle Instant Client on Linux x86-64 and Linux for Arm (aarch64)
  platforms by downloading the RPMs using the `dnf install` command.

Installing Oracle Instant Client Using Zip Files

Learn how to download and use zip files to install Oracle Instant Client on Linux x86-64 and
Linux for Arm (aarch64) platforms.

1. Go to the Oracle Instant Client Downloads page:

   https://www.oracle.com/database/technologies/instant-client/downloads.html

2. Download the desired Oracle Instant Client zip files. Select the correct platform,
   architecture, and packages of your choice. For example, if your application is 64-bit, then
   ensure that you select 64-bit Instant Client and download the Basic Package zip file.

   All installations require a Basic or Basic Light package. All other packages are optional.

3. Unzip each zip file into a single directory such as `/opt/oracle/instantclient_23_7`
   that is accessible to your application.

   For example, for **Linux x86-64**:

   ```
   $ cd /opt/oracle
   $ unzip instantclient-basic-linux.x64-23.4.0.24.05.zip
   ```

   For example, for **Linux for Arm (aarch64)**:

   ```
   $ cd /opt/oracle
   $ unzip instantclient-basic-linux.arm64-23.7.0.25.01.zip
   ```

   All the packages unzip into `/opt/oracle/instantclient_23_7`.

   > **Note:**
   >
   > The operating system may restrict runtime access to libraries unless they are
   > located in a standard directory hierarchy such as `/opt` or `/usr/local`.

4. Install the operating system `libaio` package. This is available as `libaio1` on some
   Linux distributions. For example, on Oracle Linux, run:

   ```
   $ sudo dnf install libaio
   ```

5. If Oracle Instant Client is the only Oracle software installed on your system, then update
   the runtime link path, for example:

   ```
   $ sudo sh -c "echo /opt/oracle/instantclient_23_7 > \
        /etc/ld.so.conf.d/oracle-instantclient.conf"
     $ sudo ldconfig
   ```

Alternatively, set the LD_LIBRARY_PATH environment variable prior to running applications. For example:

```
$ export LD_LIBRARY_PATH=/opt/oracle/instantclient_23_7:$LD_LIBRARY_PATH
```

Optionally, add the variable to configuration files such as `~/.bash_profile` and to application configuration files such as `/etc/sysconfig/httpd`.

6. If you intend to colocate optional Oracle configuration files such as `tnsnames.ora`, `sqlnet.ora`, `ldap.ora`, or `oraaccess.xml` with Oracle Instant Client, then move these files to the `network/admin` subdirectory.

   This is the default Oracle configuration directory for applications linked with this Oracle Instant Client.

   Alternatively, you can move the Oracle configuration files to another, accessible directory. Then set the environment variable TNS_ADMIN to that directory name.

7. To use binaries such as `sqlplus` from the SQL*Plus package, unzip the package to the same directory as the Basic package and then update your PATH environment variable. For example:

```
$ export PATH=/opt/oracle/instantclient_23_7:$PATH
```

8. Start your application.

Installing Oracle Instant Client Using RPMs

Learn how to install Oracle Instant Client on Linux x86-64 and Linux for Arm (aarch64) platforms by downloading the RPMs using the `dnf install` command.

> **Note:**
>
> Starting with 19c, by default, you can install only one version of the Oracle Instant Client RPM packages at a time. If you need multiple versions, use the Oracle Instant Client zip files.

Removing Earlier Oracle Instant Client Installations

Remove any earlier installations of Oracle Instant Client using either `dnf` or `yum` before you start a fresh installation.

1. List the currently installed Oracle Instant Client packages. For example:

   For Oracle Linux 8 and later:

```
$ sudo dnf list installed | grep instantclient
```

   The output will be like:

```
oracle-instantclient19.5-basic.x86_64 19.5.0.0.0-1 @/oracle-
instantclient19.5-basic-19.5.0.0.0-1.x86_64
```

2. Remove the Oracle Instant Client packages. For example:

For Oracle Linux 8 and later on **Linux x86-64**:

```
$ sudo dnf remove oracle-instantclient19.5-basic.x86_64
```

3. Remove the legacy Oracle Instant Client yum repository setup package. For example:

For Oracle Linux 8:

```
$ sudo dnf remove oracle-release-el8
```

For Oracle Linux 9:

```
$ sudo dnf remove oracle-release-el9
```

4. A reboot is not required.

Installing Oracle Instant Client From the yum Server

If you are using Oracle Linux and have configured your Oracle Linux `yum` server, then you can install Oracle Instant Client using the `dnf install` command.

1. Install the Oracle Linux release package to configure repository definitions for Oracle Instant Client:

For Oracle Linux 8:

```
$ sudo dnf install oracle-instantclient-release-23ai-el8
```

For Oracle Linux 9:

```
$ sudo dnf install oracle-instantclient-release-23ai-el9
```

2. Go to the Oracle Instant Client repositories to view the available Oracle Instant Client packages:

For Oracle Linux 8 on **Linux x86-64**:

https://yum.oracle.com/repo/OracleLinux/OL8/oracle/instantclient23/x86_64/index.html

For Oracle Linux 9 on **Linux x86-64**:

https://yum.oracle.com/repo/OracleLinux/OL9/oracle/instantclient23/x86_64/index.html

For Oracle Linux 8 on **Linux for Arm (aarch64)**:

https://yum.oracle.com/repo/OracleLinux/OL8/oracle/instantclient23/aarch64/index.html

All installations require a Basic or Basic Light package. All other packages are optional.

3. Install Oracle Instant Client packages using `dnf`.

For Oracle Linux 8 and later:

```
$ sudo dnf install oracle-instantclient-basic
```

4. To install other Oracle Instant Client packages like SQL*Plus, run:

For Oracle Linux 8 and later:

```
$ sudo dnf install oracle-instantclient-sqlplus
```

Using versionlock to Avoid Upgrades

To ensure that you do not inadvertently upgrade Oracle Instant Client when a new major release is published, use `dnf versionlock` after you have installed Oracle Instant Client 23ai.

```
$ sudo dnf install yum-plugin-versionlock
```

For Oracle Linux 8:

```
$ sudo dnf versionlock oracle-instantclient-release-23ai-el8
```

For Oracle Linux 9:

```
$ sudo dnf versionlock oracle-instantclient-release-23ai-el9
```

> **Note:**
>
> Configuring `dnf versionlock` still allows automatic Oracle Instant Client release updates.

Manually Downloading and Installing Oracle Instant Client RPM Packages

An alternative method for installing Oracle Instant Client is to manually download and install the Oracle Instant Client packages.

1. Go to the Oracle Instant Client Downloads page:

   https://www.oracle.com/database/technologies/instant-client/downloads.html

2. Download the desired Oracle Instant Client RPM packages. Select the correct platform, architecture, and packages of your choice. For example, if your application is 64-bit, then ensure that you select 64-bit Instant Client and download the Basic Package RPM file.

   All installations require a Basic or Basic Light package. All other packages are optional.

3. Install the packages using `dnf`. For example:

   For Oracle Linux 8 on **Linux x86-64**:

   ```
   $ sudo dnf install oracle-instantclient-basic-23.4.0.24.05-1.el8.x86_64.rpm
   ```

   For Oracle Linux 9 on **Linux x86-64**:

   ```
   $ sudo dnf install oracle-instantclient-basic-23.4.0.24.05-1.el9.x86_64.rpm
   ```

   For Oracle Linux 8 on **Linux for Arm (aarch64)**:

   ```
   $ sudo dnf install oracle-instantclient-
   basic-23.7.0.25.01-1.el8.aarch64.rpm
   ```

Additionally, you may have to perform the following tasks before you start your application:

- If you intend to colocate optional Oracle configuration files such as `tnsnames.ora`, `sqlnet.ora`, `ldap.ora`, or `oraaccess.xml` with Oracle Instant Client, then move these files to the `/usr/lib/oracle/23/client64/lib/network/admin` subdirectory.

  This is the default Oracle configuration directory for applications linked with this Oracle Instant Client.

  Alternatively, you can move the Oracle configuration files to another, accessible directory. Then set the environment variable TNS_ADMIN to that directory name.

- To use binaries from the tools package, use `yum` or `dnf` to install the package and then update your PATH environment variable. For example:

  ```
  $ export PATH=/usr/lib/oracle/23/client64/bin:$PATH
  ```

Installing Oracle Instant Client Basic Light

Learn how to install Oracle Instant Client Basic Light on Linux x86-64 and Linux for Arm (aarch64) platforms.

- About Oracle Instant Client Basic Light
  The size of the Instant Client Basic Light library is smaller than Oracle Instant Client. This is because Instant Client Basic Light contains only English language error message files, only 37 most common character set definitions out of around 250, and only monolingual collations.

- Globalization Settings for Oracle Instant Client Basic Light
  Oracle Instant Client Basic Light supports the following client character sets.

- Installing Oracle Instant Client Basic Light Packages
  Learn how to install Oracle Instant Client Basic Light by downloading zip files or RPM files, or by using the package repository.

About Oracle Instant Client Basic Light

The size of the Instant Client Basic Light library is smaller than Oracle Instant Client. This is because Instant Client Basic Light contains only English language error message files, only 37 most common character set definitions out of around 250, and only monolingual collations.

The Instant Client Basic Light version is geared toward applications that do not use legacy, rare character sets. There is no restriction on the language and the territory fields of the NLS_LANG setting, hence, Instant Client Basic Light operates with any language and territory settings supported by Oracle Database. However, the value of the NLS_SORT setting cannot be a multilingual or Unicode collation, that is, a collation with the suffix `_M` or prefix `UCA` in its name. Collations `BIG5`, `GBK`, and `HKSCS` are not supported either.

Only English error messages are provided with Instant Client Basic Light, hence, error messages generated by the client, such as Net connection errors, are always reported in English, even if you set NLS_LANG to a language other than AMERICAN. Error messages generated by the database server, such as SQL statement syntax errors, are generated in the selected language.

Globalization Settings for Oracle Instant Client Basic Light

Oracle Instant Client Basic Light supports the following client character sets.

Single-byte

- US7ASCII
- WE8DEC
- WE8ISO8859P1
- EE8ISO8859P2
- NEE8ISO8859P4
- CL8ISO8859P5
- AR8ISO8859P6
- EL8ISO8859P7
- IW8ISO8859P8
- WE8ISO8859P9
- NE8ISO8859P10
- TH8TISASCII
- VN8MSWIN1258
- WE8ISO8859P15
- BLT8ISO8859P13
- EE8MSWIN1250
- CL8MSWIN1251
- EL8MSWIN1253
- IW8MSWIN1255
- TR8MSWIN1254
- WE8MSWIN1252
- BLT8MSWIN1257
- AR8MSWIN1256

Multibyte

- JA16EUC
- JA16SJIS
- JA16EUCTILDE
- JA16SJISTILDE
- KO16MSWIN949
- ZHS16GBK
- ZHT32EUC
- ZHT16MSWIN950

ORACLE

- ZHT16HKSCS

Unicode

- UTF8

- AL32UTF8

- AL16UTF16 (only as NCHAR character set)

The preceding character sets, except AL16UTF16, can be specified in the NLS_LANG setting and can be the character set of a database to which the Instant Client Basic Light connects. The following character sets can also be the character set of a database to which the Instant Client Basic Light connects:

- WE8EBCDIC37C

- WE8EBCDIC1047

Oracle Instant Client Basic Light returns an error if any other character set is used as the client or database character set. Oracle Instant Client Basic Light can also operate with the OCI Environment handles created in the `OCI_UTF16` mode.

Installing Oracle Instant Client Basic Light Packages

Learn how to install Oracle Instant Client Basic Light by downloading zip files or RPM files, or by using the package repository.

Installing Oracle Instant Client Basic Light Using Zip Files

To download and use zip files to install Oracle Instant Client Basic Light, follow the instructions in Installing Oracle Instant Client Using Zip Files. However, unzip the `basiclite` zip file instead of the `basic` zip file.

Installing Oracle Instant Client Basic Light Using RPMs

To remove earlier installations and install Oracle Instant Client Basic Light using RPMs, follow the instructions in Installing Oracle Instant Client Using RPMs. However, use the `basiclite` packages instead of `basic`.

> **Note:**
>
> To install other packages like SQL*Plus, install the Oracle Instant Client Basic package.

Instant Client Libraries for OCI

Learn about the Oracle Database client-side files used by Oracle Call Interface (OCI) applications.

Table 6-3   Instant Client Basic Package Shared Libraries for OCI

File Name	Description
`libclntsh.so.23.1` `libclntshcore.so.23.1`	Client code libraries
`libociei.so`	OCI Instant Client Data Shared Library
`libnnz.so`	Security library
`legacy.so`	Security Provider library for legacy algorithms support
`pkcs11.so`	Security Provider library for HSM and Smartcard support
`fips.so`	Security Provider library for FIPS certified algorithms support

Table 6-4   Instant Client Basic Light Package Shared Libraries for OCI

File Name	Description
`libclntsh.so.23.1` `libclntshcore.so.23.1`	Client code libraries
`libociicus.so`	OCI Instant Client Data Shared Library
`libnnz.so`	Security Library
`legacy.so`	Security Provider library for legacy algorithms support
`pkcs11.so`	Security Provider library for HSM and Smartcard support
`fips.so`	Security Provider library for FIPS certified algorithms support

Guidelines for Basic and Basic Light Packages

- The libraries must reside in the same directory in order to use Oracle Instant Client.

- Other libraries and utilities are available when you install Oracle Instant Client, but are not needed for OCI runtime use. For example, you can delete files such as `ojdbc8.jar`.

- In general, all OCI functionality is available to applications run using Oracle Instant Client, except that the Instant Client is for client-side operations only. Therefore, server-side external procedures cannot use Oracle Instant Client libraries.

- OCI applications, by default, look for the OCI Data Shared Library, `libociei.so` in the runtime library search path, for example, in $LD_LIBRARY_PATH to determine if the application should operate in the Instant Client mode. If the OCI Data Shared Library is not found, then OCI tries to load the Oracle Instant Client Basic Light Data Shared Library `libociicus.so`.

Guidelines for Running OCI Applications

- If you have multiple directories containing Oracle Client libraries, then only one such directory should be in the operating system library search path.

- If Oracle Instant Client is the only Oracle software installed on your system, then update the runtime link path using `ldconfig`. For example:

```
$ sudo sh -c "echo /opt/oracle/instantclient_23_7 > \
    /etc/ld.so.conf.d/oracle-instantclient.conf"
$ sudo ldconfig
```

  Alternatively, to run OCI applications using Oracle Instant Client, set the operating system library search path, for example LD_LIBRARY_PATH, to the directory with the Oracle Instant Client libraries.

- If an Oracle Database installation or full Oracle Database Client installation exists on the same system, then you should not have `$ORACLE_HOME/lib` and the Oracle Instant Client directory on the operating system library search path simultaneously regardless of the order in which they appear in the operating system library search path. Either the `$ORACLE_HOME/lib` directory (for non-Instant Client operations) or Oracle Instant Client directory (for Oracle Instant Client operations) should be in the operating system library search path variable, but not both.

- If you intend to colocate optional Oracle configuration files such as `tnsnames.ora`, `sqlnet.ora`, `ldap.ora`, or `oraaccess.xml` with Oracle Instant Client, then move these files to the `network/admin` subdirectory.

Environment Variables for Oracle Instant Client

Learn about some common Oracle environment variables that can influence Oracle Instant Client installations.

Table 6-5   Common Environment Variables for Oracle Instant Client

Environment Variable	Description	For More Information
LD_LIBRARY_PATH	Used on Linux and some UNIX platforms. Set this to include the Oracle Instant Client libraries, for example `$ORACLE_HOME/lib` or `/opt/oracle/instantclient_23_7`. Not needed if the libraries are located by an alternative method, such as from running `ldconfig`. On other platforms you will need to set the OS specific equivalent, such as LIBPATH or SHLIB_PATH. **Note**: You should not set this environment variable if you have installed the Instant Client RPMs, or if the libraries are located by an alternative method, such as by running `ldconfig`.	*Oracle Database Administrator's Reference*
NLS_LANG	If necessary, set the NLS_LANG environment variable. The NLS_LANG environment variable sets the language and territory used by the client application and the database server. It also sets the client's character set, which is the character set for data entered or displayed by a client program. If not set, a default value will be chosen by Oracle Instant Client.	*Oracle Database Globalization Support Guide*

Table 6-5   (Cont.) Common Environment Variables for Oracle Instant Client

Environment Variable	Description	For More Information
TNS_ADMIN	The location of the optional Oracle Net configuration files and Oracle Instant Client configuration files, including `tnsnames.ora`, `sqlnet.ora`, and `oraaccess.xml`, if they are not in the default location.	*Oracle Database Net Services Administrator's Guide*
	If TNS_ADMIN is not set, then the `instantclient_23_7/network/admin` subdirectory should contain your Oracle Net Services configuration files.	
ORA_TZFILE	Oracle Instant Client can use the ORA_TZFILE environment variable to read the time zone file from the file system when this environment variable is set.	*Oracle Database Administrator's Reference*
	If ORA_TZFILE is not set, then Oracle Instant Client uses the larger, default, `timezlrg_n.dat` file from the shared libraries. If you want Oracle Instant Client to use the smaller `timezone_n.dat` file, then set the ORA_TZFILE environment variable to the name of the file without any absolute or relative path names.	
	For example:	
	`$ export ORA_TZFILE=timezone_n.dat`	
	Where, `n` is the time zone data file version number.	
	Run the `genezi -v` command to determine the client library timezone information.	
	To use an external time zone file, create an `oracore/zoneinfo` subdirectory under the Oracle Instant Client directory, and move the time zone file into this `oracore/zoneinfo` subdirectory. Now, set ORA_TZFILE to the time zone file name, without any absolute or relative path names.	
	With Oracle Instant Client 19.18 or later, you can alternatively place the external time zone file in any directory, and then set the ORA_TZFILE environment variable to the absolute path of the file. For example:	
	`$ export ORA_TZFILE=/opt/oracle/config/timezone_n.dat`	
ORA_SDTZ	Specifies the default session time zone.	*Oracle Database Globalization Support Guide*
ORA_SUPPRESS_ERROR_URL	Setting this environment variable to TRUE disables the error help URL generation in applications that call the OCIErrorGet() function.	*Oracle Call Interface Developer's Guide*

> **Note:**
>
> Do not set the ORACLE_HOME environment variable for Oracle Instant Client installations.

About the Oracle Instant Client SDK

Learn about the Software Development Kit (SDK) development tools for Oracle Instant Client.

The SDK is a set of development tools that you can use to create applications for Oracle Instant Client. The SDK is available for download as an RPM or a zip file for Oracle Linux x86-64 and Arm platforms. Here are some examples of these RPM and zip files:

For Linux x86-64 (Oracle Linux 8)

- RPM: `oracle-instantclient-devel-23.4.0.24.05-1.el8.x86_64.rpm`
- Zip file: `instantclient-sdk-linux.x64-23.4.0.24.05.zip`

For Linux x86-64 (Oracle Linux 9)

- RPM: `oracle-instantclient-devel-23.4.0.24.05-1.el9.x86_64.rpm`
- Zip file: `instantclient-sdk-linux.x64-23.4.0.24.05.zip`

For Linux for Arm (aarch64)

- RPM: `oracle-instantclient-devel-23.7.0.25.01-1.el8.aarch64.rpm`
- Zip file: `instantclient-sdk-linux.arm64-23.7.0.25.01.zip`

The SDK contains the following:

- Both C and C++ header files and a `makefile` for developing OCI and OCCI applications while in an Oracle Instant Client environment. Developed applications can be deployed in any client environment.
- Both C and C++ demonstration programs.
- The Object Type Translator (OTT) utility and its classes to generate application header files.
- On Linux, the `demo.mk makefile` is included to build demos. For example, the `instantclient_23_7` directory must be in the runtime library search path, and set LD_LIBRARY_PATH before linking the application.

Patching Oracle Instant Client Shared Libraries

Learn how to patch Oracle Instant Client shared libraries with a complete Oracle Database Client installation.

You must rebuild your Oracle Instant Client packages and libraries as part of the patching process.

- Patching Oracle Instant Client
  Perform the following steps to patch Oracle Instant Client.

- **About Rebuilding Oracle Instant Client Packages and Libraries**
  Separate `makefile` targets are used to create the data shared libraries, zip files, and RPM files either individually or all together.

- **Regenerating Data Shared Libraries**
  Learn how to regenerate the Oracle Instant Client data shared libraries `libociei.so` and `libociicus.so`.

- **Regenerating Zip Files and RPM Files**
  Learn how to regenerate the zip files and RPM files for the various Oracle Instant Client packages.

Patching Oracle Instant Client

Perform the following steps to patch Oracle Instant Client.

1. Complete a full Oracle Database Client Administration installation based on Oracle home for patching.

2. Apply the patch to your full Oracle Database Client Administration installation using the `OPatch` utility.

3. Regenerate the data shared libraries, zip files, and RPM files.

4. Copy the data shared libraries, zip files, and RPM files to the target system and unzip them to the Oracle Instant Client directory.

The `OPatch` utility stores the patching information of the ORACLE_HOME installation in `libclntsh.so`. You can retrieve this patching information using the `genezi -v` command.

If the Oracle Instant Client deployment system does not have the `genezi` utility, copy it from the `$ORACLE_HOME/bin` directory.

Related Topics

- **Instant Client Libraries for OCI**
  Learn about the Oracle Database client-side files used by Oracle Call Interface (OCI) applications.

- Patching Using OPatch

About Rebuilding Oracle Instant Client Packages and Libraries

Separate `makefile` targets are used to create the data shared libraries, zip files, and RPM files either individually or all together.

Regenerating the data shared libraries requires both a compiler and a linker, which may not be available on all installations.

The regenerated Oracle Instant Client binaries contain only the Oracle Instant Client files installed in the Oracle Client Administrator Home from which you regenerate the libraries and files. Hence, error messages, character set encodings, and time zone files that are available in the regeneration environment are the only ones that are packaged in the data shared libraries. Error messages, character set encodings, and time zone files depend on which national languages were selected for the installation of the Oracle Client Administrator Home.

> **✎ Note:**
>
> Regeneration of data shared libraries, zip files, and RPM files is not available on Microsoft Windows.

Regenerating Data Shared Libraries

Learn how to regenerate the Oracle Instant Client data shared libraries `libociei.so` and `libociicus.so`.

You can regenerate the Oracle Call Interface (OCI) Oracle Instant Client data shared libraries `libociei.so` and `libociicus.so` using the following commands in an Administrator install of a full Oracle Database Client:

1. Set ORACLE_HOME to the software install directory. For example:

   ```
   $ export ORACLE_HOME=/home/user/product/23.0.0/client_1/
   ```

2. Set LD_LIBRARY_PATH to the library directory. For example:

   ```
   $ export LD_LIBRARY_PATH=$ORACLE_HOME/lib
   ```

3. Create the following directory:

   ```
   $ mkdir -p $ORACLE_HOME/rdbms/install/instantclient/light
   ```

4. Under Oracle home directory, go to the `/rdbms/lib` directory.

   ```
   $ cd $ORACLE_HOME/rdbms/lib
   ```

5. To regenerate both `libociei.so` and `libociicus.so`:

   ```
   $ make -f ins_rdbms.mk igenliboci
   ```

Regenerating Zip Files and RPM Files

Learn how to regenerate the zip files and RPM files for the various Oracle Instant Client packages.

Before you regenerate the zip files and RPM files, install the `rpm-build` package.

For Oracle Linux 8 and later:

```
$ sudo dnf install rpm-build
```

ORACLE

Table 6-6   Commands to Regenerate the Zip Files and RPM Files

Package Name	Commands
All Packages	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_all_zip`
Basic	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_basic_zip`
Basic Light	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_basiclite_zip`
JDBC	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_jdbc_zip`
ODBC	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_odbc_zip`
SQL*Plus	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_sqlplus_zip`
Tools	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_tools_zip`
SDK	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_sdk_zip`
Precompilers	`$ cd $ORACLE_HOME/rdbms/lib` `$ make -f ins_rdbms.mk ic_precomp_zip`

The new zip files and RPM files are generated under the `$ORACLE_HOME/rdbms/install/instantclient` directory.

7
Oracle Database Client Postinstallation Tasks

Complete configuration tasks after you install Oracle Database.

You are required to complete some configuration tasks after Oracle Database Client is installed. In addition, Oracle recommends that you complete additional tasks immediately after installation. You must also complete product-specific configuration tasks before you use those products.

> **Note:**
>
> This chapter describes basic configuration only. Refer to product-specific administration and tuning guides for more detailed configuration and tuning information.

- Required Postinstallation Tasks
  Download and apply required patches for your software release after completing your initial installation.
- Recommended Postinstallation Tasks
  Oracle recommends that you complete these tasks after installation.

Required Postinstallation Tasks

Download and apply required patches for your software release after completing your initial installation.

- Downloading Release Update Patches
  Download and install Release Updates (RU) and Monthly Recommended Patches (MRPs) patches for your Oracle software after you complete your installation.

Downloading Release Update Patches

Download and install Release Updates (RU) and Monthly Recommended Patches (MRPs) patches for your Oracle software after you complete your installation.

Oracle provides quarterly updates in the form of Release Updates (RU) and Monthly Recommended Patches (MRPs). Oracle no longer releases patch sets. For more information, see My Oracle Support Note 2285040.1.

Check the My Oracle Support website for required updates for your installation.

1. Log in to the My Oracle Support website:

   https://support.oracle.com

> **Note:**
>
> If you are not a My Oracle Support registered user, then click **Register for My Oracle Support** and register.

2. In the **sign in to other Oracle Support portals** section, click **Support for Oracle Hardware, Software, and Managed Cloud services** to log in to My Oracle Support.

3. On the main My Oracle Support page, click **Patches & Updates**.

4. In the **Patch Search** region, select **Product or Family (Advanced)**.

5. On the **Product or Family (Advanced)** display, provide information about the product, release, and platform for which you want to obtain patches, and click **Search**.

   The Patch Search pane opens, displaying the results of your search.

6. Select the patch number and click **ReadMe**.

   The README page is displayed. It contains information about the patch and how to apply the patches to your installation.

7. Uncompress the Oracle patch updates that you downloaded from My Oracle Support.

> **Note:**
>
> The use of OPatch and OPatchAuto for out-of-place patching continues to be deprecated.
> For patching in Oracle Database 23ai, Oracle recommends that you use out-of-place patching using Gold images to apply quarterly release updates (RUs). This deprecation does not affect in-place patching. If you want to perform in-place patching, then OPatch and OPatchAuto continue to be available for this purpose. The Oracle Database 23ai Free release is not supported for patching with RUs.

Related Topics

- My Oracle Support note 888.1
- Patch Delivery Methods for Oracle Database

Recommended Postinstallation Tasks

Oracle recommends that you complete these tasks after installation.

- Setting Language and Locale Preferences for Client Connections
  Configure client applications connecting to an Oracle Database according to your locale preferences and your I/O device character set.

Setting Language and Locale Preferences for Client Connections

Configure client applications connecting to an Oracle Database according to your locale preferences and your I/O device character set.

You must configure client applications connecting to an Oracle Database according to your locale preferences and your I/O device character set. If your applications do not have their own specific methods to configure locale preferences, then the method you use to configure an

Oracle database client connection depends on the access API you use to connect to the database. Check your application documentation, before you configure locale preferences for your applications.

For applications that connect to Oracle Databases using Oracle Call Interface (OCI) use NLS_LANG and other client settings with names that start with `NLS_` to set the locale conventions and client character set for Oracle Database sessions. It is important that you set the character set part of the NLS_LANG value properly. The Oracle character set whose name you set in NLS_LANG must correspond to the character encoding used by your I/O devices. For Linux, it is the character encoding of the terminal or terminal emulator, and usually corresponds to what the `locale` command reports. For Microsoft Windows, it is either the ANSI Code Page (for GUI applications), such as WE8MSWIN1252, or the OEM Code Page (for Console mode applications), such as US8PC437. By doing this, the OCI API is notified about the character set of data that it receives from the application. OCI can then convert this data correctly to and from the database character set.

You can specify NLS_LANG and the other NLS settings as environment variables. On Microsoft Windows, you can also specify them as Registry settings. Environment variable values take precedence over Registry values.

Oracle Universal Installer sets a default value for the NLS_LANG setting in Registry when it creates a new Oracle home on Microsoft Windows. The NLS_LANG value is based on the language of the Windows user interface, which is the language of Windows menu items and dialog box labels. The installer does not set NLS_LANG on Linux and other UNIX system-based operating systems.

> ⚠️ **Caution:**
>
> Failure to set the client character set correctly can cause data loss.

Java applications that connect to Oracle Databases by using Oracle JDBC do not use NLS_LANG. Instead, Oracle JDBC maps the default locale of the Java VM in which the application runs to the Oracle Database language and territory settings. Oracle JDBC then configures the connected database session using these settings. Because Java works internally in Unicode, the client character set is always set to Unicode. Unless an application explicitly changes it, the default locale of the Java VM is set based on the locale of the user operating system on which the Java VM runs. Check your Java VM documentation for information about configuring the Java VM default locale.

> ✎ **Note:**
>
> In 3-tier architecture deployments, application servers that are database clients can have settings in their configuration files that specify the NLS_LANG value or the Java VM locale. Check the documentation accompanying these servers.

> ✎ **See Also:**
>
> *Oracle Database Globalization Support Guide* for more information about configuring user locale preferences

8
Removing Oracle Database Software

Use the `deinstall` command that is included in Oracle homes to remove Oracle software. Oracle does not support the removal of individual products or components.

> ⚠ **Caution:**
>
> If you have a single-instance database on a node in a cluster, and if you have multiple databases with the same global database name (GDN), then you cannot use the `deinstall` command to remove one database only.

- About Oracle Deinstallation Options
  You can stop and remove Oracle Database software and components in an Oracle Database home with the `deinstall` command.

- Oracle Deinstallation (Deinstall)
  You can run the `deinstall` command from an Oracle home directory after installation.

- Deinstallation Examples for Oracle Database Client
  Use these examples to help you understand how to run the `deinstall` command.

About Oracle Deinstallation Options

You can stop and remove Oracle Database software and components in an Oracle Database home with the `deinstall` command.

You can remove the following software using `deinstall`:

- Oracle Database

- Oracle Grid Infrastructure, which includes Oracle Clusterware and Oracle Automatic Storage Management (Oracle ASM)

- Oracle Real Application Clusters (Oracle RAC)

- Oracle Database Client

The `deinstall` command is available in Oracle home directories after installation. It is located in the `$ORACLE_HOME/deinstall` directory.

`deinstall` creates a response file by using information in the Oracle home and using the information you provide. You can use a response file that you generated previously by running the `deinstall` command using the `-checkonly` option. You can also edit the response file template.

If you run `deinstall` to remove an Oracle Grid Infrastructure installation, then the deinstaller prompts you to run the `deinstall` command as the `root` user. For Oracle Grid Infrastructure for a cluster, the script is `rootcrs.sh`, and for Oracle Grid Infrastructure for a standalone server (Oracle Restart), the script is `roothas.sh`.

> **✎ Note:**
>
> You must run the `deinstall` command from the same release to remove Oracle software. Do not run the `deinstall` command from a later release to remove Oracle software from an earlier release. For example, do not run the `deinstall` command from the 23ai Oracle home to remove Oracle software from an existing 19c Oracle home.

If the software in the Oracle home is not running (for example, after an unsuccessful installation), then `deinstall` cannot determine the configuration, and you must provide all the configuration details either interactively or in a response file.

In addition, before you run `deinstall` for Oracle Grid Infrastructure installations:

- Dismount Oracle Advanced Cluster File System (Oracle ACFS) and disable Oracle Automatic Storage Management Dynamic Volume Manager (Oracle ADVM).

- If Grid Naming Service (GNS) is in use, then notify your DNS administrator to delete the subdomain entry from the DNS.

Additional Files Deleted by deinstall

When you run `deinstall`, if the central inventory (`oraInventory`) contains no other registered homes besides the home that you are deconfiguring and removing, then `deinstall` removes the following files and directory contents in the Oracle base directory of the Oracle Database installation owner:

- `admin`

- `cfgtoollogs`

- `checkpoints`

- `diag`

- `oradata`

- `fast_recovery_area`

Oracle strongly recommends that you configure your installations using an Optimal Flexible Architecture (OFA) configuration, and that you reserve Oracle base and Oracle home paths for exclusive use of Oracle software. If you have any user data in these locations in the Oracle base that is owned by the user account that owns the Oracle software, then `deinstall` deletes this data.

> **⚠ Caution:**
>
> - `deinstall` deletes Oracle Database configuration files, user data, and fast recovery area (FRA) files even if they are located outside of the Oracle base directory path.
>
> - Do not manually remove files and directories in these locations as `deinstall` and the installer manage these directories. Be careful before placing files in these locations.

Oracle Deinstallation (Deinstall)

You can run the `deinstall` command from an Oracle home directory after installation.

Purpose

`deinstall` stops Oracle software, and removes Oracle software and configuration files on the operating system for a specific Oracle home.

Syntax

The `deinstall` command uses the following syntax:

```
(./deinstall [-silent] [-checkonly] [-paramfile complete path of input
response file]
[-params name1=value name2=value . . .]
[-o complete path of directory for saving files]
[-tmpdir complete path of temporary directory to use]
[-logdir complete path of log directory to use] [-help]
```

Parameters

Parameter	Description
`-silent`	Use this flag to run `deinstall` in noninteractive mode. This option requires one of the following:
	• A working system that it can access to determine the installation and configuration information. The `-silent` flag does not work with failed installations.
	• A response file that contains the configuration values for the Oracle home that is being deinstalled or deconfigured.
	You can generate a response file to use or modify by running the `deinstall` command with the `-checkonly` flag. `deinstall` then discovers information from the Oracle home to deinstall and deconfigure. It generates the response file that you can then use with the `-silent` option.
	You can also modify the template file `deinstall.rsp.tmpl`, located in the `$ORACLE_HOME/deinstall/response` directory.
`-checkonly`	Use this flag to check the status of the Oracle software home configuration. Running `deinstall` with the `-checkonly` flag does not remove the Oracle configuration. The `-checkonly` flag generates a response file that you can use with the `deinstall` command and `-silent` option.

Parameter	Description
`-paramfile` *complete path of input response file*	Use this flag to run `deinstall` with a response file in a location other than the default. When you use this flag, provide the complete path where the response file is located.
	The default location of the response file is `$ORACLE_HOME/deinstall/response.`
`-params` [*name1=value name2=value name3=value . . .*]	Use this flag with a response file to override one or more values to change in a response file you have created.
`-o` *complete path of directory for saving response files*	Use this flag to provide a path other than the default location where the response file (`deinstall.rsp.tmpl`) is saved.
	The default location of the response file is `$ORACLE_HOME/deinstall/response.`
`-tmpdir` *complete path of temporary directory to use*	Use this flag to specify a non-default location where `deinstall` writes the temporary files for the deinstallation.
`-logdir` *complete path of log directory to use*	Use this flag to specify a non-default location where `deinstall` writes the log files for the deinstallation.
`-local`	Use this flag on a multinode environment to deinstall Oracle software in a cluster.
	When you run `deinstall` with this flag, it deconfigures and deinstalls the Oracle software on the local node (the node where `deinstall` is run). On remote nodes, it deconfigures Oracle software, but does not deinstall the Oracle software.
`-help`	Use this option to obtain additional information about the command option flags.

Deinstallation Examples for Oracle Database Client

Use these examples to help you understand how to run the `deinstall` command.

You can run `deinstall` from the `$ORACLE_HOME/deinstall` directory. The deinstallation starts without prompting you for the Oracle home path.

```
$ ./deinstall
```

You can generate a deinstallation response file by running the `deinstall` command with the `-checkonly` flag.

```
./deinstall -silent --checkonly
```

The `deinstall` tool generates a response file in `/tmp` and displays the location. For example, `/tmp/deinstall2024-08-01_01-40-28PM/response/deinstall_Ora23Client1.rsp`

Use the optional flag `-paramfile` to run this response file:

```
$ cd /u01/app/oracle/product/23.0.0/client_1/deinstall
$ ./deinstall -paramfile /tmp/deinstall2024-08-01_01-40-28PM/response/
deinstall_Ora23Client1.rsp
```

A
Response Files

Review the following topics to install and configure Oracle Database, Oracle Grid Infrastructure, and other Oracle products using response files.

- Modes for Oracle DBCA Noninteractive (Silent) Configuration
  You run the Oracle DBCA installer from a command prompt using either silent mode or response file mode.

- Reasons for Using Silent Mode or Response File Mode
  Review this section for use cases for running the installer in silent mode or response file mode.

- Using Response Files
  Use response files to install and configure Oracle products using the installer in silent or response file mode.

- Preparing Response Files
  Review this information to prepare response files for use during silent mode or response file mode installations.

- Running Oracle Universal Installer Using a Response File
  After creating the response file, run Oracle Univeral Installer at the command line, specifying the response file you created, to perform the installation.

Modes for Oracle DBCA Noninteractive (Silent) Configuration

You run the Oracle DBCA installer from a command prompt using either silent mode or response file mode.

When you start Oracle Universal Installer (OUI), you can use a response file to automate the installation and configuration of Oracle software, either fully or partially. OUI uses the values contained in the response file to provide answers to some or all installation prompts.

Typically, the installer runs in interactive mode, which means that it prompts you to provide information in graphical user interface (GUI) screens. When you use response files to provide this information, you run the installer from a command prompt using either of the following modes:

- Silent mode

  If you include responses for all of the prompts in the response file and specify the `-silent` option when starting the installer, then it runs in silent mode. During a silent mode installation, the installer does not display any screens. Instead, it displays progress information in the terminal that you used to start it.

- Response file mode

  If you include responses for some or all of the prompts in the response file and omit the `-silent` option, then the installer runs in response file mode. During a response file mode installation, the installer displays all the screens, screens for which you specify information in the response file, and also screens for which you did not specify the required information in the response file.

You define the settings for a silent or response file installation by entering values for the variables listed in the response file. For example, to specify the Oracle home name, provide the Oracle home path for the `ORACLE_HOME` environment variable:

```
ORACLE_HOME=/u01/app/oracle/product/23.0.0/dbhome_1
```

Reasons for Using Silent Mode or Response File Mode

Review this section for use cases for running the installer in silent mode or response file mode.

Mode	Uses
Silent	Use silent mode for the following installations: • Complete an unattended installation, which you schedule using operating system utilities such as `at`. • Complete several similar installations on multiple systems without user interaction. • Install the software on a system that does not have X Window System software installed on it. The installer displays progress information on the terminal that you used to start it, but it does not display any of the installer screens.
Response file	Use response file mode to complete similar Oracle software installations on more than one system, providing default answers to some, but not all of the installer prompts.

Using Response Files

Use response files to install and configure Oracle products using the installer in silent or response file mode.

> **Note:**
>
> You must complete all required preinstallation tasks on a system before running the installer in silent or response file mode.

1. Prepare a response file.

2. Run the installer in silent or response file mode.

3. Run the root scripts as prompted by Oracle Universal Installer.

4. If you completed a software-only installation, then run Net Configuration Assistant and Oracle DBCA in silent or response file mode to create the database listener and an Oracle Database instance respectively.

Preparing Response Files

Review this information to prepare response files for use during silent mode or response file mode installations.

- Editing a Response File Template
  For Oracle Database Client, response files are located in the `$ORACLE_HOME/response` directory.

- Recording Response Files
  You can use Oracle Universal Installer (OUI) in interactive mode to record response files, which you can then edit and use to complete silent mode or response file mode installations.

Editing a Response File Template

For Oracle Database Client, response files are located in the `$ORACLE_HOME/response` directory.

All response file templates contain comment entries, sample formats, examples, and other useful instructions. Please read these instructions as they help you specify values for the variables listed in the response files and customize your installation.

The following table lists the response files provided with this software:

Table A-1   Response Files for Oracle Database Client

Response File	Description
`client_install.rsp`	Silent installation of Oracle Database Client
`netca.rsp`	Silent configuration of Oracle Net using Oracle Net Configuration Assistant.

> **Caution:**
>
> When you modify a response file template and save a file for use, the response file may contain plain text passwords. Ownership of the response file should be given to the Oracle software installation owner only, and permissions on the response file should be changed to 600. Oracle strongly recommends that database administrators or other administrators delete or secure response files when they are not in use.

To copy and modify a response file:

1. Copy the response file from the response file directory to a directory on your system:

   ```
   $ cp /directory_path/inventory/response/response_file.rsp local_directory
   ```

   In this example, `directory_path` is the path of the directory where you have copied the installation binaries.

2. Open the response file in a text editor:

   ```
   $ vi /local_dir/response_file.rsp
   ```

3. Follow the instructions in the file to edit it.

> **Note:**
>
> The installer or configuration assistant fails if you do not correctly configure the response file. Also, ensure that your response file name has the `.rsp` suffix.

4. Secure the response file by changing the permissions on the file to 600:

```
$ chmod 600 /local_dir/response_file.rsp
```

Ensure that only the Oracle software owner user can view or modify response files or consider deleting them after the installation succeeds.

> **Note:**
>
> A fully-specified response file for an Oracle Database Client installation contains the passwords for database administrative accounts and for a user who is a member of the OSDBA group (required for automated backups).

Recording Response Files

You can use Oracle Universal Installer (OUI) in interactive mode to record response files, which you can then edit and use to complete silent mode or response file mode installations.

You can save all the installation steps into a response file during installation by clicking **Save Response File** on the Summary page. You can use the generated response file for a silent installation later.

When you record the response file, you can either complete the installation, or you can exit from the installer on the Summary page, before OUI starts to set up the software to the system.

If you use record mode during a response file mode installation, then the installer records the variable values that were specified in the original source response file into the new response file.

> **Note:**
>
> OUI does not save passwords while recording the response file.

To record a response file:

1. Complete preinstallation tasks for an Oracle Database Client installation.

   When you run the installer to record a response file, it checks the system to verify that it meets the requirements to install the software. For this reason, Oracle recommends that you complete all of the required preinstallation tasks and record the response file while completing an installation.

2. Ensure that the Oracle software owner user (typically `oracle`) has permissions to create or write to the Oracle home path that you will specify when you run the installer.

3. On each installation screen, specify the required information.

4. When OUI displays the Summary screen, perform the following steps:

ORACLE

a. Click **Save Response File**. In the window, specify a file name and location for the new response file. Click Save to write the responses you entered to the response file.

b. Click **Finish** to continue with the installation.

Click **Cancel** if you do not want to continue with the installation. The installation stops, but the recorded response file is retained.

> ✏️ **Note:**
>
> Ensure that your response file name has the `.rsp` suffix.

5. Before you use the saved response file on another system, edit the file and make any required changes. Use the instructions in the file as a guide when editing it.

Running Oracle Universal Installer Using a Response File

After creating the response file, run Oracle Univeral Installer at the command line, specifying the response file you created, to perform the installation.

Run Oracle Universal Installer at the command line, specifying the response file you created. The Oracle Universal Installer executable, `runInstaller`, provides several options. For help information on the full set of these options, run the `runInstaller` command with the `-help` option. For example:

```
$ directory_path/runInstaller -help
```

The help information appears in a window after some time.

To run the installer using a response file:

1. Complete the preinstallation tasks as for a normal installation

2. Log in as the software installation owner user.

3. If you are completing a response file mode installation, then set the operating system `DISPLAY` environment variable for the user running the installation.

> ✏️ **Note:**
>
> You do not have to set the `DISPLAY` environment variable if you are completing a silent mode installation.

4. To start the installer in silent or response file mode, enter a command similar to the following:

```
$ /directory_path/runInstaller [-silent] [-noconfig] \
 -responseFile responsefilename
```

> ✏️ **Note:**
>
> Do not specify a relative path to the response file. If you specify a relative path, then the installer fails.

In this example:

- *directory_path*is the path of the directory where you have copied the installation binaries.

- `-silent` runs the installer in silent mode.

- `-noconfig` suppresses running the configuration assistants during installation, and a software-only installation is performed instead.

- *responsefilename* is the full path and file name of the installation response file that you configured.

5. If this is the first time you are installing Oracle software on your system, then Oracle Universal Installer prompts you to run the orainstRoot.sh script.

   Log in as the `root` user and run the `orainstRoot.sh` script:

   ```
   $ su root
   password:
   # /u01/app/oraInventory/orainstRoot.sh
   ```

   > **Note:**
   >
   > You do not have to manually create the `oraInst.loc` file. Running the `orainstRoot.sh` script is sufficient as it specifies the location of the Oracle Inventory directory.

6. When the installation completes, log in as the `root` user and run the `root.sh` script. For example

   ```
   $ su root
   password:
   # /oracle_home_path/root.sh
   ```