

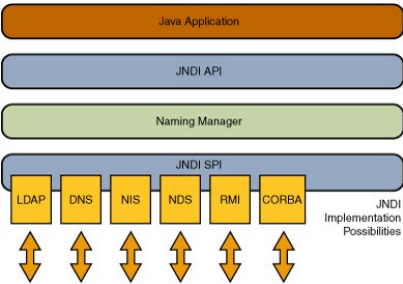
*The Java Tutorials have been written for JDK 8. Examples and practices described in this page don't take advantage of improvements introduced in later releases and might use technology no longer available. See [Java Language Changes](#) for a summary of updated language features in Java SE 9 and subsequent releases. See [JDK Release Notes](#) for information about new features, enhancements, and removed or deprecated options for all JDK releases.*

Lesson: Overview of JNDI

The Java Naming and Directory Interface™ (JNDI) is an application programming interface (API) that provides [naming](#) and [directory](#) functionality to applications written using the Java™ programming language. It is defined to be independent of any specific directory service implementation. Thus a variety of directories -new, emerging, and already deployed can be accessed in a common way.

Architecture

The JNDI architecture consists of an API and a service provider interface (SPI). Java applications use the JNDI API to access a variety of naming and directory services. The SPI enables a variety of naming and directory services to be plugged in transparently, thereby allowing the Java application using the JNDI API to access their services. See the following figure:



Packaging

JNDI is included in the Java SE Platform. To use the JNDI, you must have the JNDI classes and one or more service providers. The JDK includes service providers for the following naming/directory services:

- Lightweight Directory Access Protocol (LDAP)
- Common Object Request Broker Architecture (CORBA) Common Object Services (COS) name service
- Java Remote Method Invocation (RMI) Registry
- Domain Name Service (DNS)

Other service providers can be downloaded from the [JNDI page](#) or obtained from other vendors.

The JNDI is divided into five packages:

- [javax.naming](#)
- [javax.naming.directory](#)
- [javax.naming.ldap](#)
- [javax.naming.event](#)
- [javax.naming.spi](#)

The next part of the lesson has a brief description of the JNDI packages.