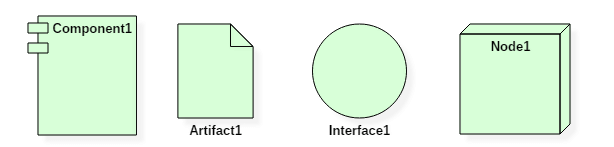
**Purpose of a deployment diagram**

Deployment diagrams are used with the sole purpose of describing how software is deployed into the hardware system. It visualizes how software interacts with the hardware to execute the complete functionality. It is used to describe software to hardware interaction and vice versa.

**Deployment Diagram Symbol and notations**

Deployment Diagram Notations

A deployment diagram consists of the following notations:

1. A node
2. A component
3. An artifact
4. An interface

**What is an artifact?**

An artifact represents the specification of a concrete real-world entity related to software development. You can use the artifact to describe a framework which is used during the software development process or an executable file. Artifacts are deployed on the nodes. The most common artifacts are as follows,

1. Source files
2. Executable files
3. Database tables
4. Scripts
5. DLL files
6. User manuals or documentation
7. Output files

Artifacts are deployed on the nodes. It can provide physical manifestation for any UML element. Generally, they manifest components. Artifacts are labeled with the stereotype **<<artifact>>,**and it may have an artifact icon on the top right corner.

Each artifact has a filename in its specification that indicates the physical location of the artifact. An artifact can contain another artifact. It may be dependent on one another.

Artifacts have their properties and behavior that manipulates them.

Generally, an artifact is represented as follows in the unified modeling language.

artifact

**Node**

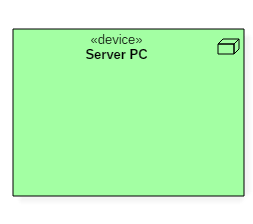
Node is an essential UML element that describes the execution of code and the communication between various entities of a system.

Generally, a node has two stereotypes as follows:

* **<< device >>**It is a node that represents a physical machine capable of performing computations. A device can be a router or a server PC. It is represented using a node with stereotype <<device>>.

In the UML model, you can also nest one or more devices within each other.

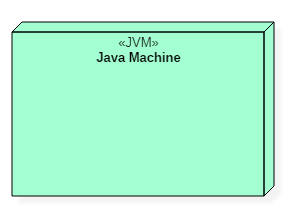
Following is a representation of a device in UML:



device node

* **<< execution environment >>**It is a node that represents an environment in which software is going to execute. For example, Java applications are executed in java virtual machine (JVM). JVM is considered as an execution environment for Java applications. We can nest an execution environment into a device node. You can net more than one execution environments in a single device node.

Following is a representation of an execution environment in UML:



execution environment node

**Example:**

