

SOFTWARE ENGINEERING PRACTICAL DOCUMENT

PRACTICAL NO: 10

AIM: Study and implementation of Deployment Diagrams.

What is Deployment Diagram?

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

The deployment diagram maps the software architecture created in design to the physical system architecture that executes it. In distributed systems, it models the distribution of the software across the physical nodes.

Purpose of Deployment Diagram

The main purpose of the deployment diagram is to represent how software is installed on the hardware component. It depicts in what manner a software interacts with hardware to perform its execution.

Both the deployment diagram and the component diagram are closely interrelated to each other as they focus on software and hardware components. The component diagram represents the components of a system, whereas the deployment diagram describes how they are actually deployed on the hardware.

Deployment Diagram Symbol and notations

A deployment diagram consists of the following notations:

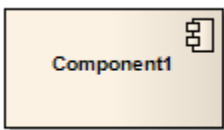
1. A node
2. A component
3. An artifact

4. An interface

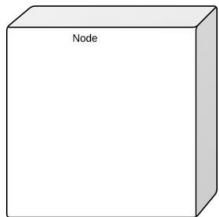
Artifact: A product developed by the software, symbolized by a rectangle with the name and the word “artifact” enclosed by double arrows. An artifact represents the specification of a concrete real-world entity related to software development. Artifacts are deployed on the nodes. The most common artifacts are Source files, Executable files, Database tables, Scripts, DLL files, User manuals or documentation, Output files



Component: A rectangle with two tabs that indicates a software element.



Node: A hardware or software object, shown by a three-dimensional box. A node is a physical thing that can execute one or more artifacts. An association between nodes represents a communication path from which information is exchanged in any direction.

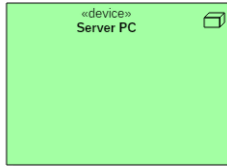


Generally, a node has two stereotypes as follows:

<< 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>>.

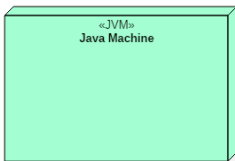
In the UML model, you can also nest one or more devices within each other.

Following is a representation of a device in UML:



<< 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:



Association: A line that indicates a message or other type of communication between nodes.

Dependency: A dashed line that ends in an arrow, which indicates that one node or component is dependent on another.

How to Draw a Deployment Diagram

Follow the simple steps below to draw a deployment diagram. You can either use the deployment diagram examples below to get a head start or use our UML diagram tool to start from the beginning.

Step 1: Identify the purpose of your deployment diagram. And to do so, you need to identify the nodes and devices within the system you'll be visualizing with the diagram.

Step 2: Figure out the relationships between the nodes and devices. Once you know how they are connected, proceed to add the communication associations to the diagram.

Step 3: Identify what other elements like components, active objects you need to add to complete the diagram.

Step 4: Add dependencies between components and objects as required.

Example of Deployment Diagram (Library Management System)

The deployment diagram shows the scenario when the system is deployed. It has 4 nodes represented with boxes and components within. The Library Management System node has the component of several databases such as books, borrowers, and users. Then the librarian must be connected to the network thru TCP/IP in order for them to access the system.

Additionally, the software is connected to an ISP which enables it to pass data to the online server and then will be accessed by the borrowers thru browsers also with the help of URLs. Lastly, the Librarian and the Borrowers can communicate with the use of ISP.

