



UNIVERSITY OF ENGINEERING AND TECHNOLOGY, TAXILA

FACULTY OF TELECOMMUNICATION AND INFORMATION ENGINEERING

SOFTWARE ENGINEERING DEPARTMENT

Lab # 09

Working with Deployment Diagram

You'll learn in this Lab:

- What a deployment diagram is
- Applying deployment diagrams
- Deployment diagrams in the big picture of the UML

A solid blueprint for setting up the hardware is essential to system design. The UML provides you with symbols for creating a clear picture of how the final hardware setup should look, along with the items that reside on the hardware

What is a Deployment Diagram?

A deployment diagram shows how artifacts are deployed on system hardware, and how the pieces of hardware connect to one another. The main hardware item is a **node**, a generic name for a computing resource.

A deployment diagram in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("**nodes**") exist (e.g., a web server, an application server, and a database server), what software components ("**artifacts**") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

In UML 2.0 a cube represents a node (as was the case in UML 1.x). You supply a name for the node, and you can add the keyword «Device», although it's usually not necessary.

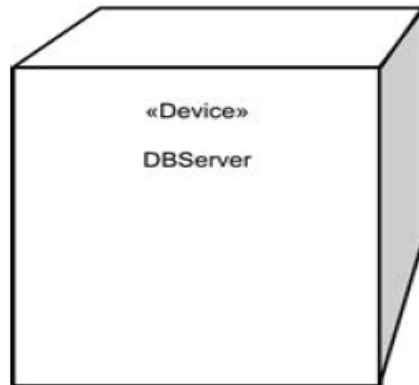


Figure1. Representing a node in the UML

A Node is either a hardware or software element. It is shown as a three-dimensional box shape, as shown below.

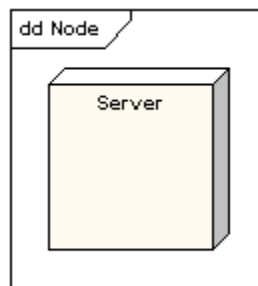


Figure2. Representing a node in the UML

Node Instance:

A node instance can be shown on a diagram. An instance can be distinguished from a node by the fact that its name is underlined and has a colon before its base node type. An instance may or may not have a name before the colon. The following diagram shows a named instance of a computer.

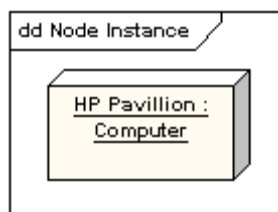


Figure3. Representing a node instance

Artifact:

An artifact is a product of the software development process. That may include process models (e.g. use case models, design models etc), source files, executables, design documents, test reports, prototypes, user manuals, etc.

An artifact is denoted by a rectangle showing the artifact name, the «artifact» keyword and a document icon, as shown below.

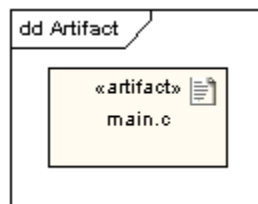


Figure4. Representing an artifact

Association:

In the context of a deployment diagram, an association represents a communication path between nodes. The following diagram shows a deployment diagram for a network, depicting network protocols as stereotypes, and multiplicities at the association ends.

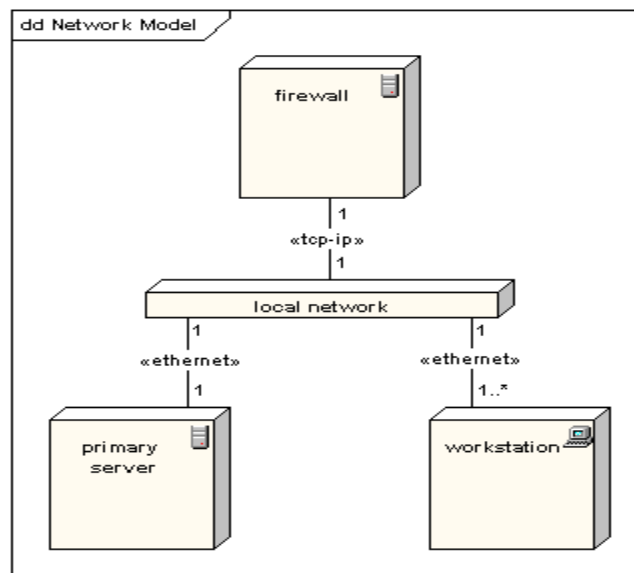


Figure5. Representing association between nodes

Node as Container:

A node can contain other elements, such as components or artifacts. The following diagram shows a deployment diagram for part of an embedded system, depicting an executable artifact as being contained by the motherboard node.

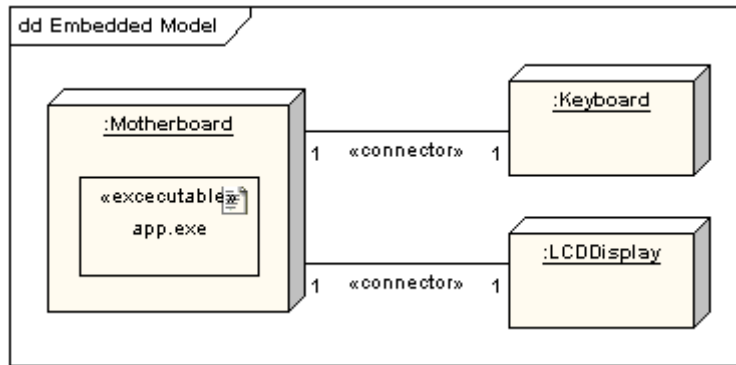


Figure6. Embedded deployment model

Ways of representing Deployment Diagram:

The figure 7 shows the three ways of representing deployment diagram.

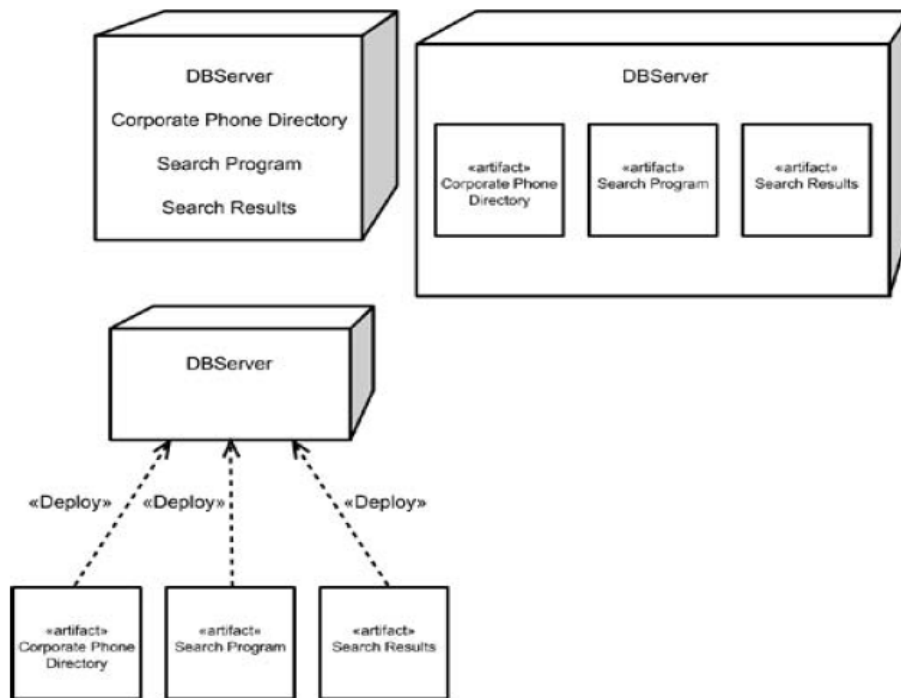


Figure7. Three ways to model the deployment of artifacts on a node

The line joining two cubes represents a connection between two nodes. Bear in mind that a connection isn't necessarily a piece or cable. You can also represent a wireless connection through it.

Android Application Deployment – Example:

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Android relies on Linux OS for core system services such as security, memory management, process management, network stack, and driver model. The Linux kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

Android applications are written in Java. Android SDK tools compile and package the code along with any required data and resource files into Android application **archive file** having .apk suffix. The .apk file represents one Android application to be deployed to the Android-enabled mobile devices.

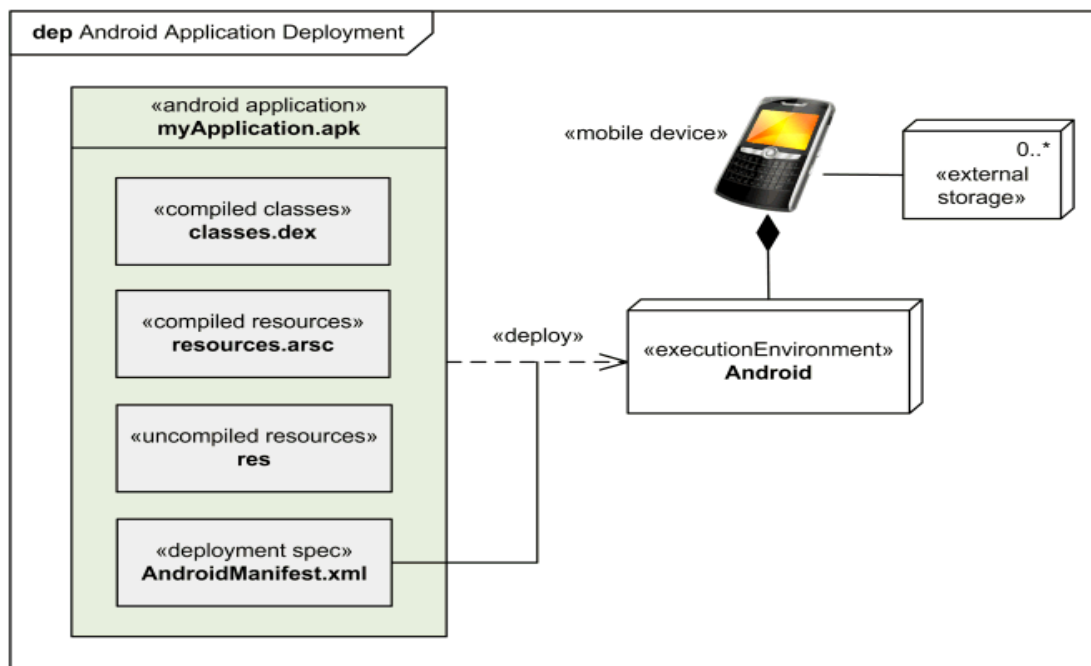


Figure8. Android Application Deployment

Modeling with Deployment Diagram

The following elements are available in a deployment diagram.

- Package
- Node
- NodeInstance

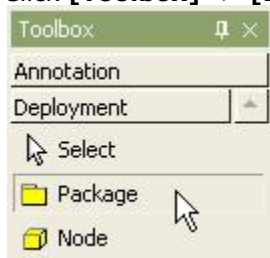
- Artifact
- Port
- Part
- Association
- DirectedAssociation
- Dependency
- Link
- Connector

Package

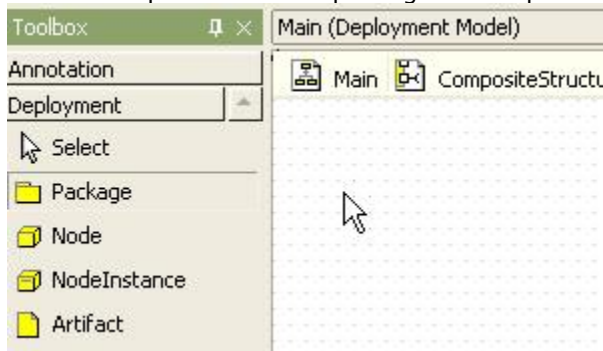
Procedure for creating package

In order to create Package in deployment diagram,

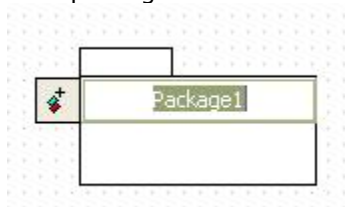
1. Click **[Toolbox] -> [Deployment] -> [Package]** button.



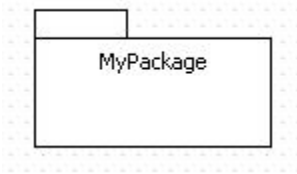
2. Click at the position where package will be placed in the **[main window]**.



3. Then package will be created.



4. At the quick dialog, enter package name and press **[Enter]** key. Then procedure is done.



Node

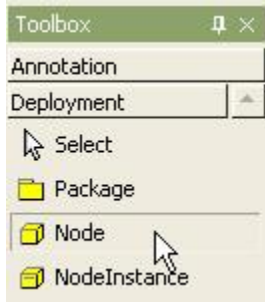
Semantics:

A node is a run-time physical object that represents a computational resource, generally having at least a memory and often processing capability as well, and upon which components may be deployed.

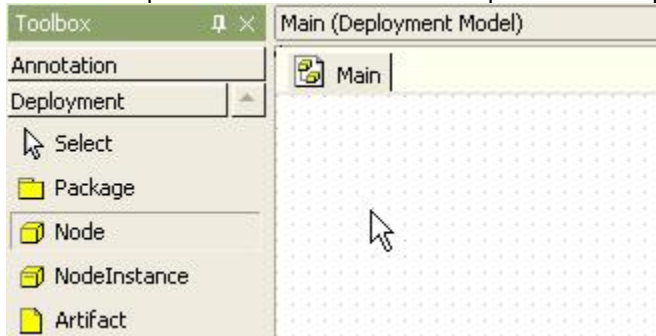
Procedure for creating node

In order to create Node in deployment diagram,

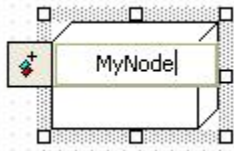
1. Click **[Toolbox] -> [Deployment] -> [Node]** button.



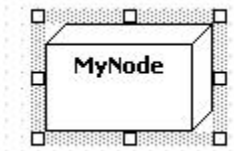
2. Click at the position where Node will be placed in the **[main window]**.



3. Then node is created and the quick dialog appears. Enter the node name at the quick dialog.



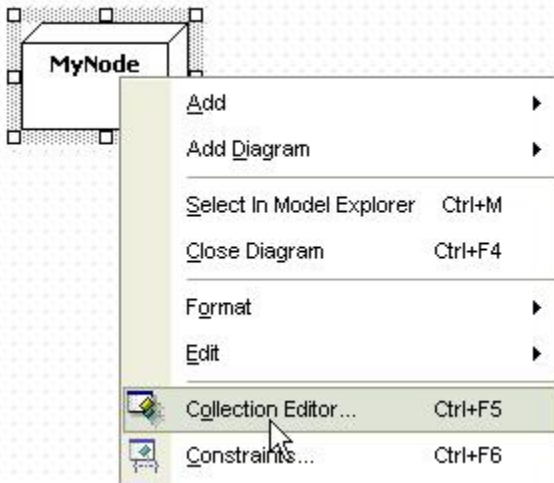
4. And press **[Enter]** key.



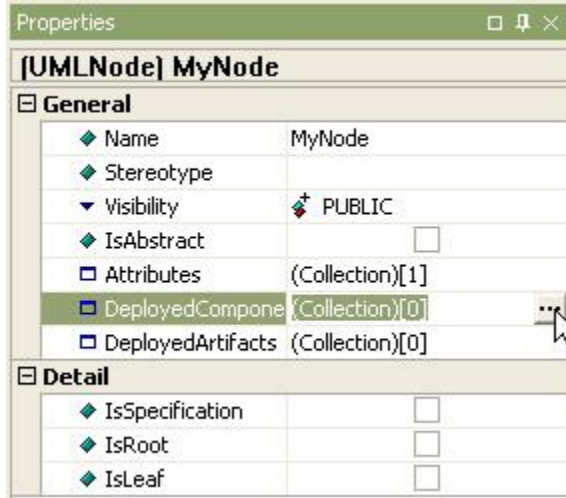
Procedure for adding deployed component


In order to add deployed component to node

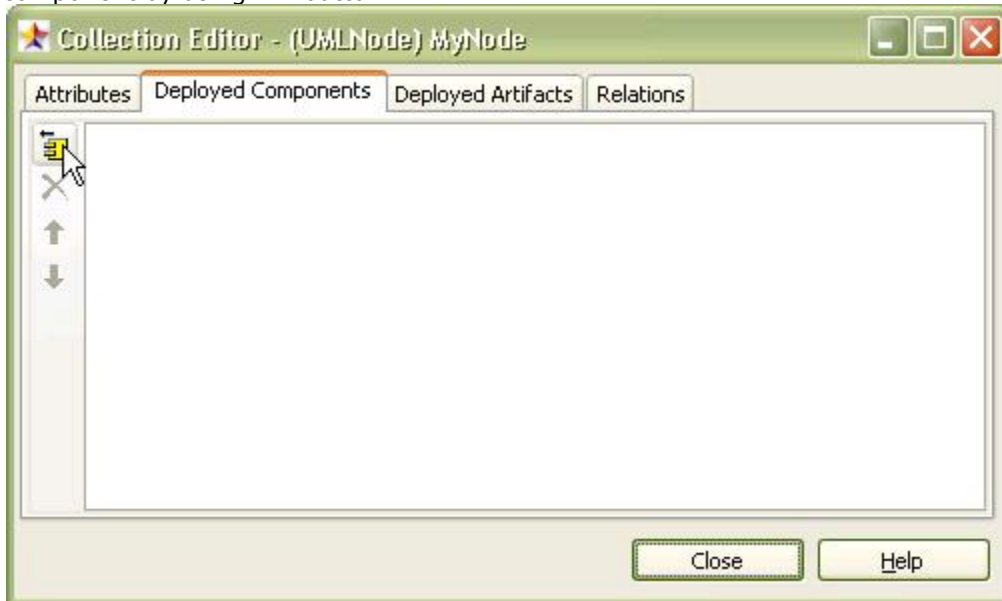
1. Select **[Collection Editor...]** popup menu of node.



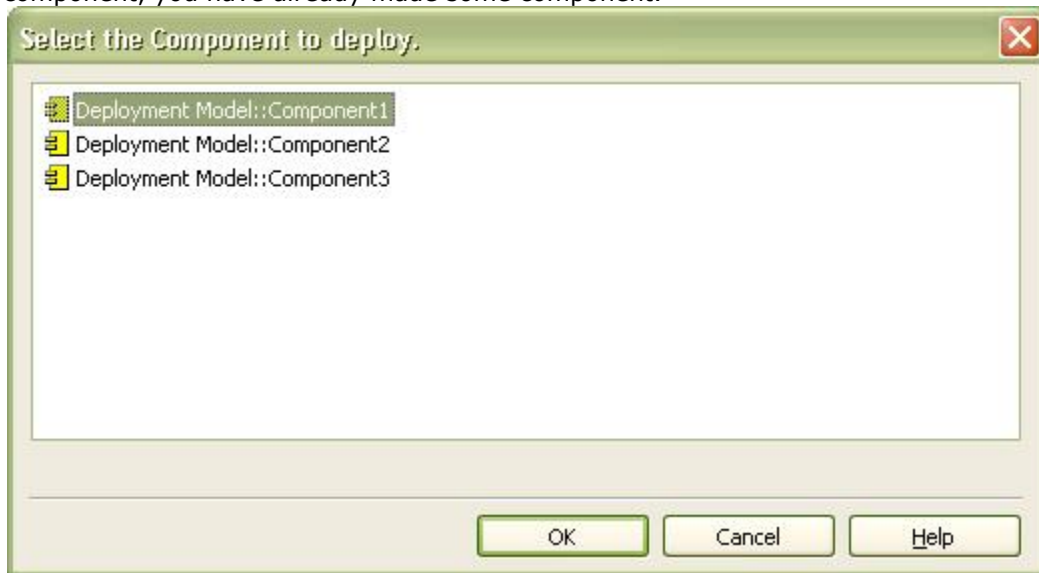
2. Or click  button in **[DeployedComponents]** property on properties window.



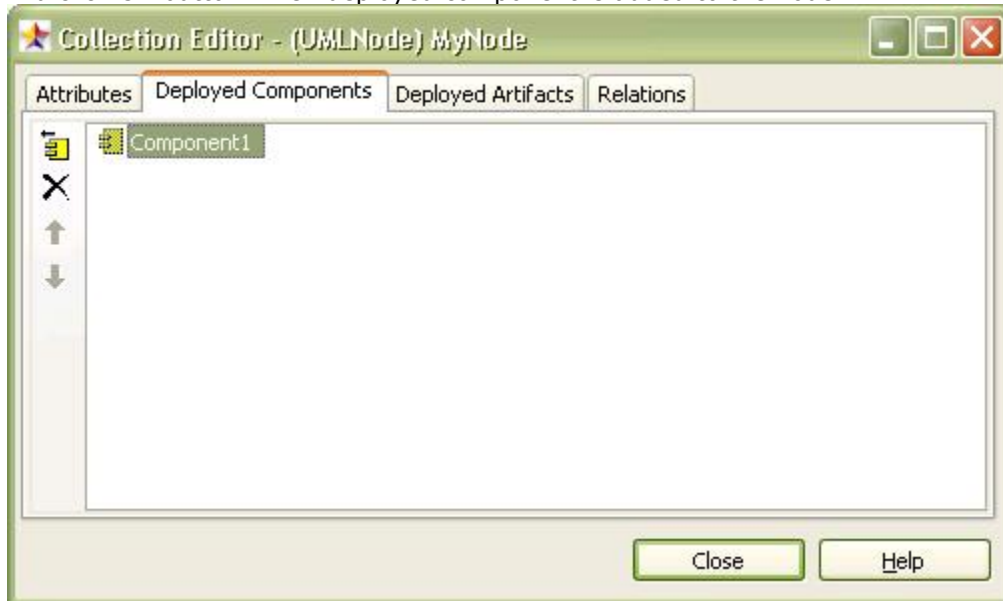
3. At the **[Deployed Components]** tab of the **[collection editor]**, you can add deployed component by using  button.



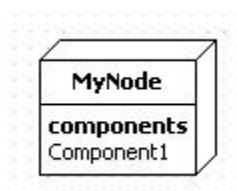
- At **[Select the Component to deploy]** dialog, select deployed component. To select component, you have already made some component.



- And click OK button. Then deployed component is added to the node.



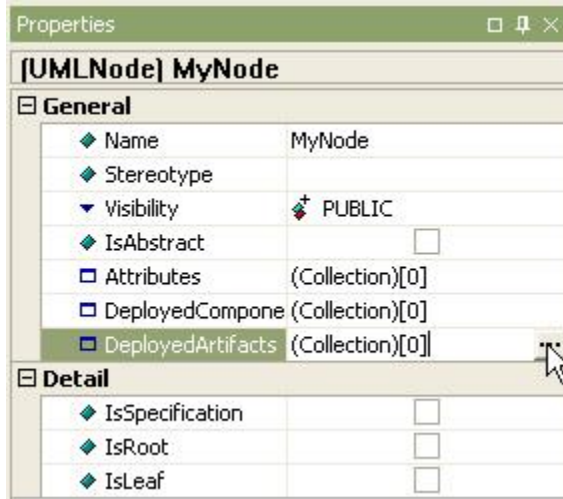
- The node is shown as following.



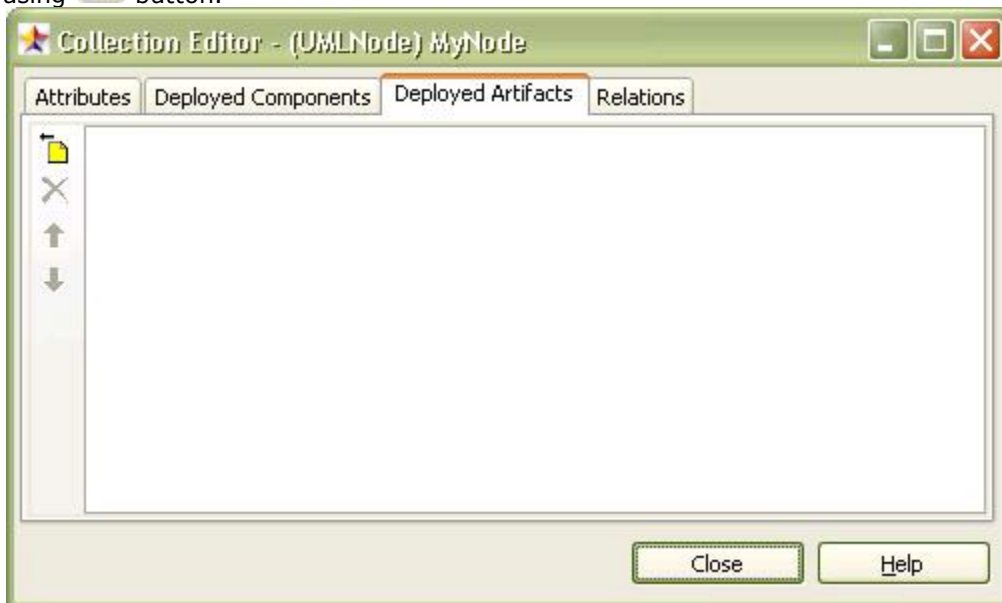
Procedure for adding deployed artifact

In order to add deployed artifact to node,

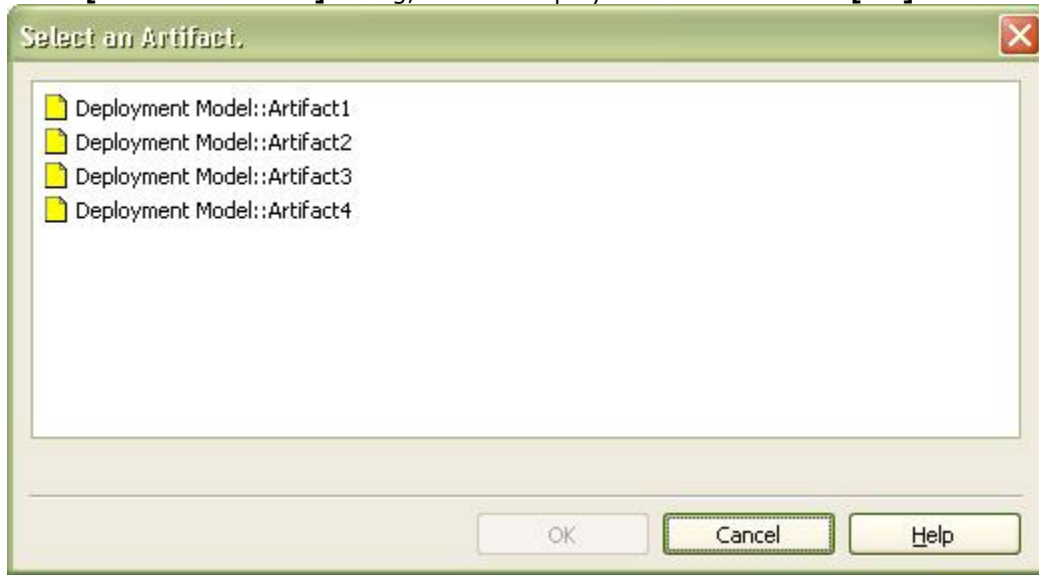
1. Select [**Collection Editor...**] popup menu of node.
2. Or click **...** button in [**DeployedArtifacts**] property on properties window.



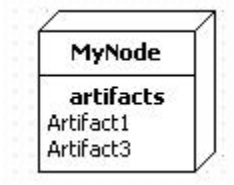
3. At the [**Deployed Artifacts**] tab of the [**collection editor**], you can add deployed artifact by using  button.



4. At the **[Select a Artifact]** dialog, select a deployed artifact and click **[OK]** button.



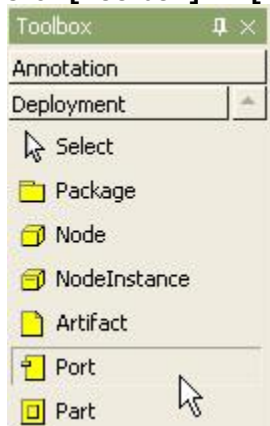
5. Then the artifact is add to the node and the node is shown as following.



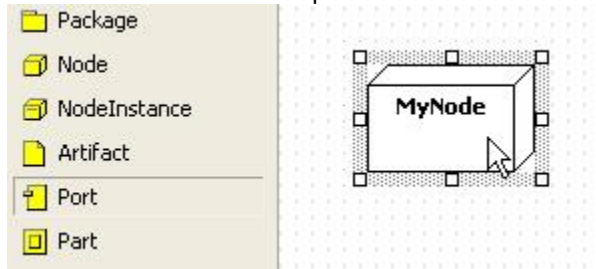
Procedure for creating port

In order to create port on a node,

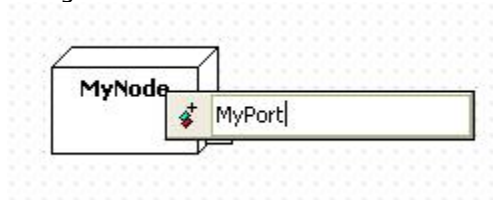
1. Click **[Toolbox]** -> **[Deployment]** -> **[Port]** button.



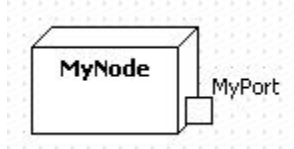
2. Click the node where the port will be contained in the **[main window]**.



3. A port is created on the node and the quick dialog appears. Enter the port name at the quick dialog.



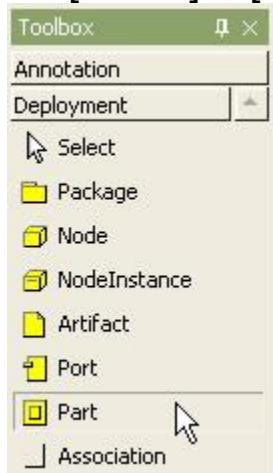
4. And press **[Enter]** key. The result is like the following.



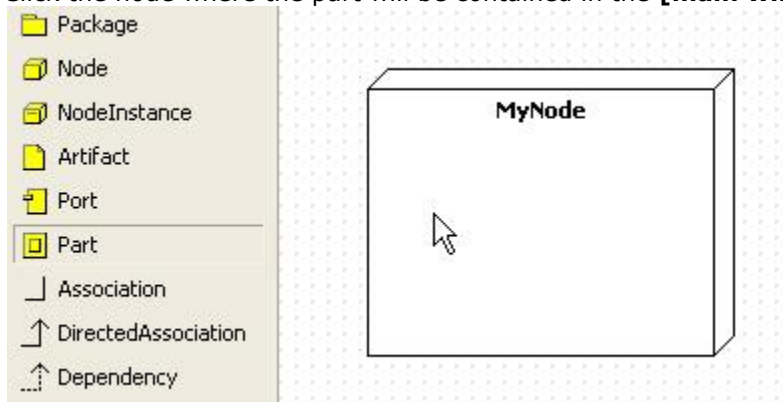
Procedure for creating part

In order to create part on a node

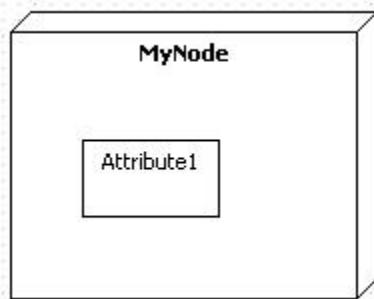
1. Click **[Toolbox] -> [Deployment] -> [Part]** button.



2. Click the node where the part will be contained in the **[main window]**.



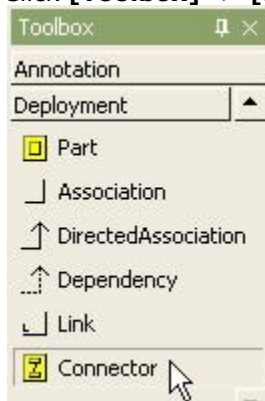
3. Finally, a part is created on the node as following.



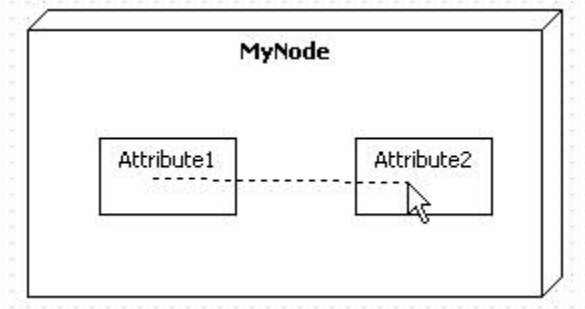
Procedure for creating connector

In order to create connector between two parts,

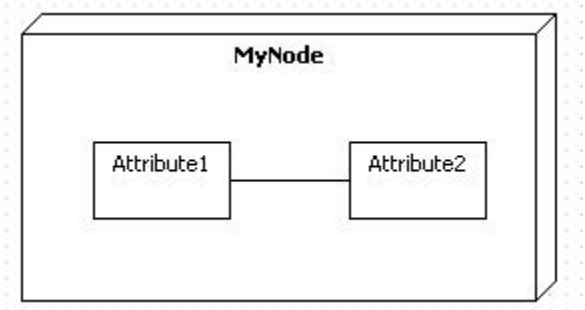
1. Click **[Toolbox] -> [Deployment] -> [Connector]** button.



2. Drag from one part and drop to the other part in the **[main window]**.



3. The result is as follows.



NodeInstance

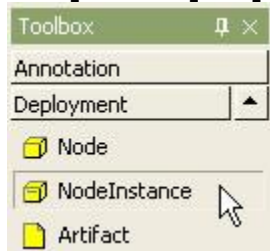
Semantics

A node instance is an instance of a node. A collection of component instances may reside on the node instance.

Procedure for creating node instance

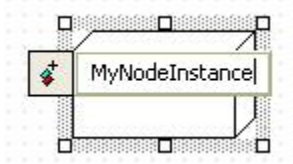
In order to create NodeInstance in deployment diagram,

1. Click **[Toolbox] -> [Deployment] -> [NodeInstance]** button.

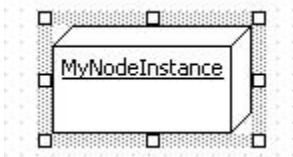


2. Click at the position where NodeInstance will be placed in the **[main window]**, a node is created, and quick dialog appears.

3. Enter the node instance name at the quick dialog and press **[Enter]** key.



4. The result is as follows.



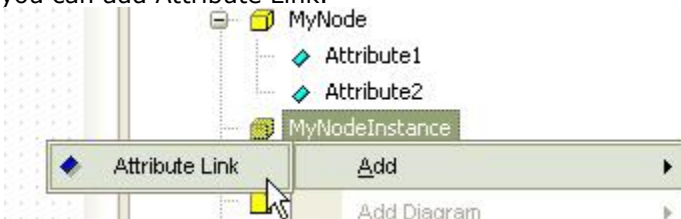
Procedure for adding attribute link to node instance

There are two way to add attribute link to node instance.

- using NodeInstance model in the **[main window]** or the **[model explorer]**
- using **[collection editor]**

In the case of using NodeInstance model

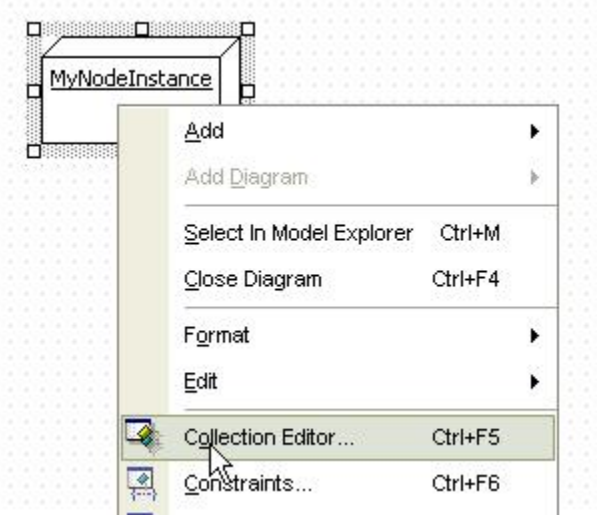
1. Select NodeInstance in the **[main window]** or in the **[model explorer]**,
2. Right-click the selected NodeInstance, select **[Add] -> [Attribute Link]** popup menu, and you can add Attribute Link.



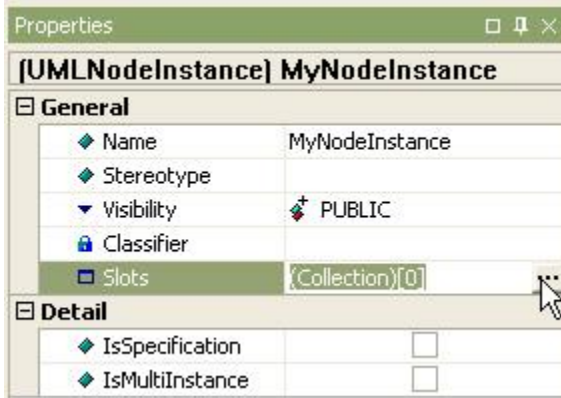
3. The node doesn't show attribute link on the view.

In the other case

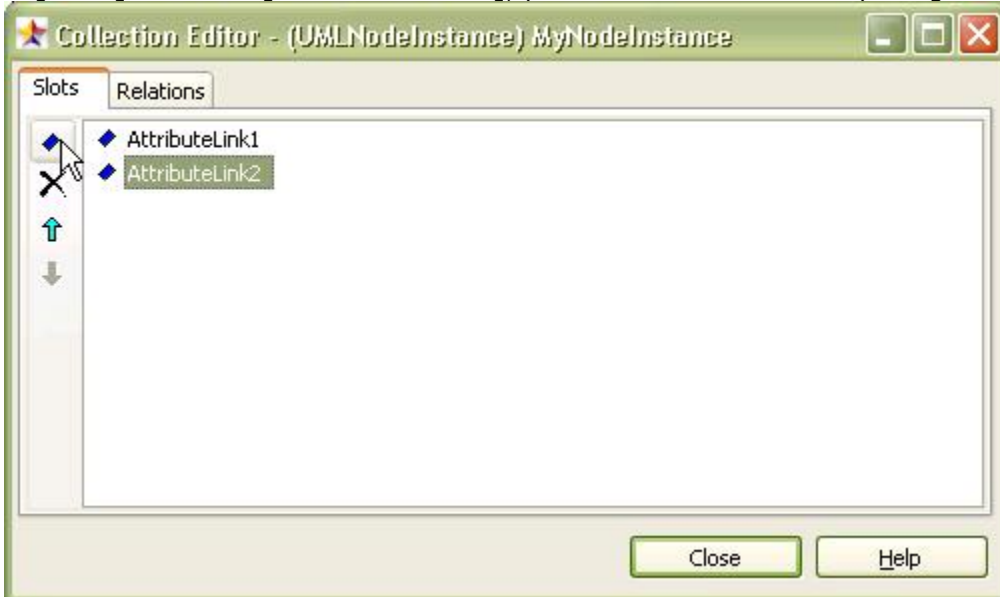
1. Select [**Collection Editor...**] popup menu of NodeInstance.



2. Or click [...] button in [**Slots**] property on properties window.



3. At [**Slots**] tab of the [**collection editor**], you can add attribute link by using [] button.

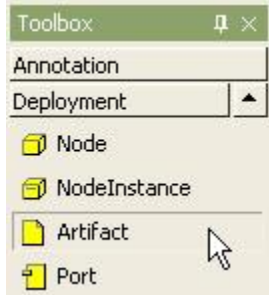


Artifact

Procedure for creating artifact

In order to create Artifact,

1. Click **[Toolbox] -> [Deployment] -> [Artifact]** button.



2. Click at the position where Artifact will be placed in the **[main window]**.
3. At the quick dialog, enter the artifact name and press **[Enter]** key.
4. The result is as follows.

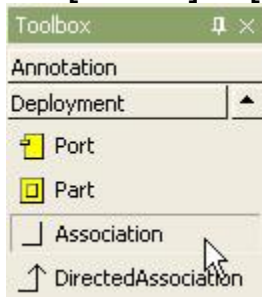


Association

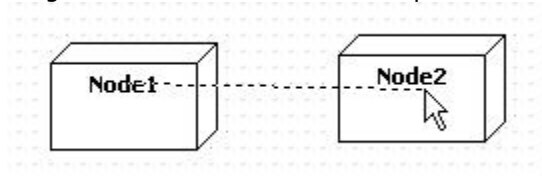
Procedure for creating association

In order to create association,

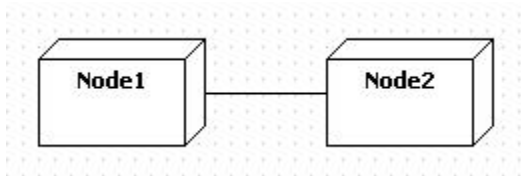
1. Click **[Toolbox] -> [Deployment] -> [Association]** button.



2. Drag from one associated and drop to another in the **[main window]**.



3. The result is as follows.

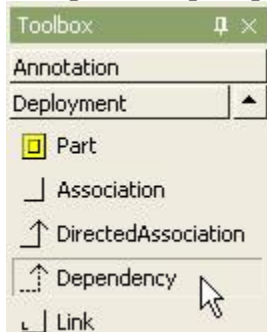


Dependency

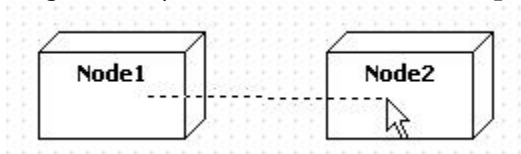
Procedure for creating dependency

In order to create dependency,

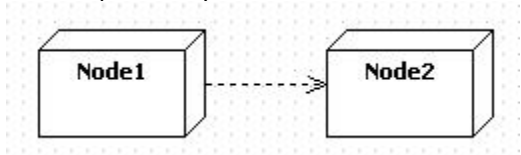
1. Click **[Toolbox] -> [Deployment] -> [Dependency]** button.



2. Drag and drop between elements in the **[main window]** in depending direction.



3. Then dependency between two elements is created as follows.

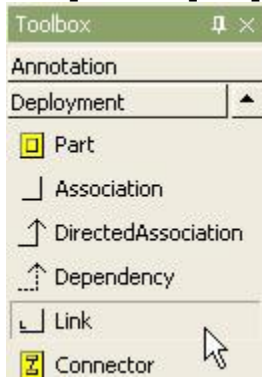


Link

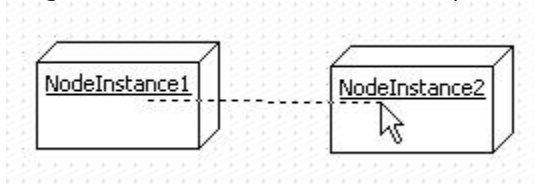
Procedure for creating link

In order to create Link,

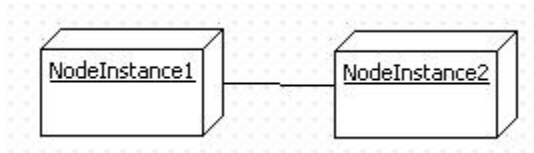
1. Click **[Toolbox]** -> **[Deployment]** -> **[Link]** button.



2. Drag from one NodeInstance and drop to the other NodeInstance in the **[main window]**.



3. Then the link between two node instances is created.



Lab Task:

Activity 1:

Draw a Deployment Diagram for Course Registration System:

- Create a new Deployment diagram with name “CRS-Deployment”.
- Create 3 new Nodes: “ClientBrowser”, “WebServer”, “DBServer”.
- Create communication path: ClientBrowser - WebServer and WebServer - DBServer
- Create an artifact “CourseRegistrationSystemUI” and <<deploy>> the artifact to “ClientBrowser” node.
- Create an artifact “CourseRegistrationSystemControl” and <<deploy>> the artifact to “WebServer” node.
- Create an artifact “CourseRegistrationSystemDB” and <<deploy>> the artifact to “DBServer” node.

Activity 2:

The *iTunes setup* application could be downloaded from *iTunes website* and installed on a home computer. After installation and registration iTunes application could communicate with Apple *iTunes Store*. Customer can buy and download music, video, TV shows, audio books, etc. and store it in *media library*. *Mobile devices* like Apple iPod Touch and Apple iPhone could update own *media libraries* from home computer with iTunes through USB, or could download media directly from Apple *iTunes Store* using some *wireless protocol* like Wi-Fi, 3G, or EDGE.

Draw an Apple iTunes Deployment Diagram.
