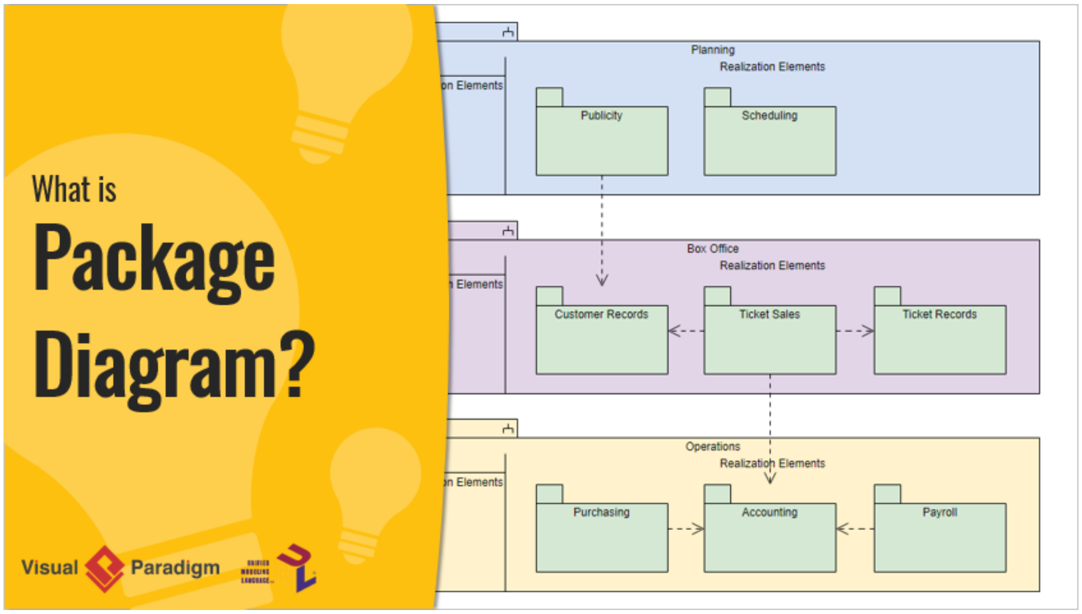
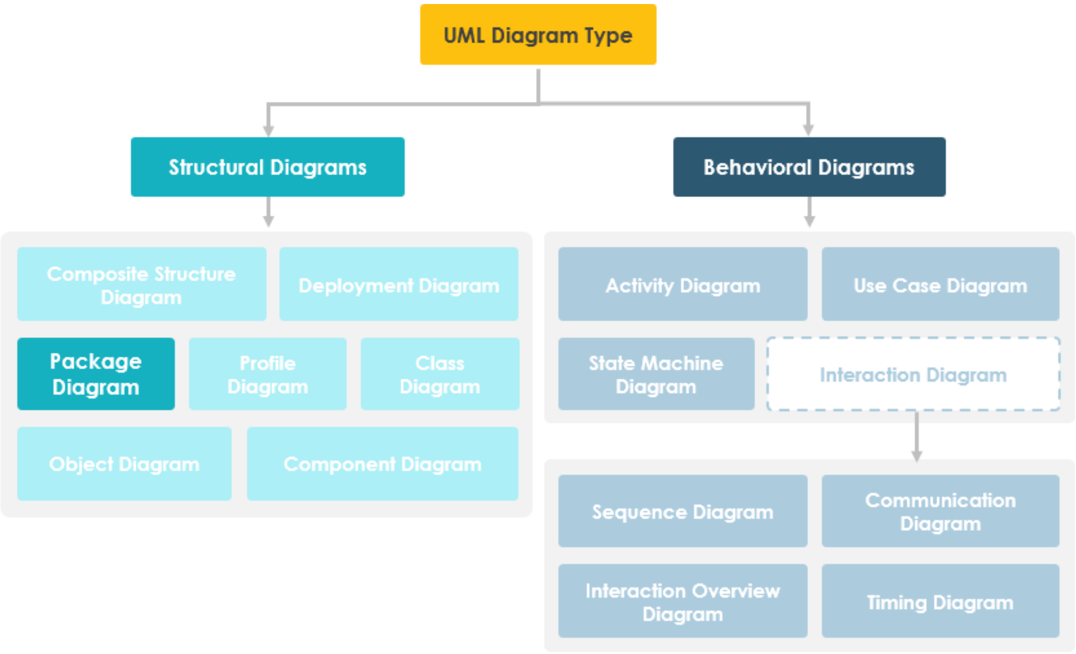
What is Package Diagram?



Package diagram, a kind of structural diagram, shows the arrangement and organization of model elements in middle to large scale project. Package diagram can show both structure and dependencies between sub-systems or modules, showing different views of a system, for example, as multi-layered (aka multi-tiered) application - multi-layered application model.

Related Links



Purpose of Package Diagrams

Package diagrams are used to structure high level system elements. Packages are used for organizing large system which contains diagrams, documents and other key deliverables.

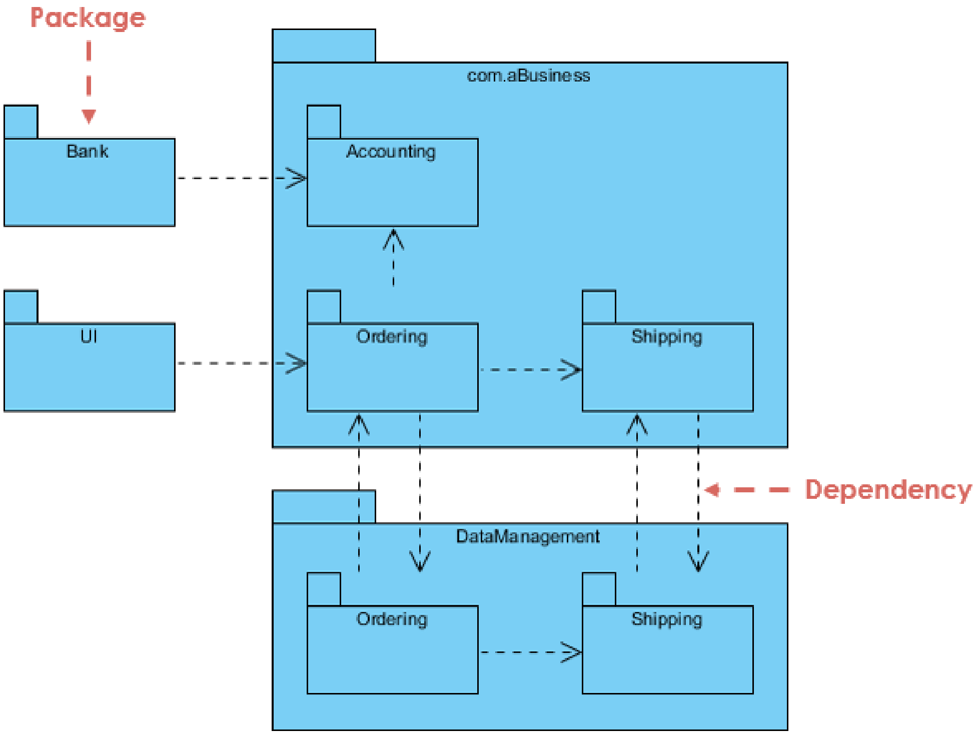
- Package Diagram can be used to simplify complex class diagrams, it can group classes into packages.
- A package is a collection of logically related UML elements.
- Packages are depicted as file folders and can be used on any of the UML diagrams.

Package Diagram at a Glance

Package diagram is used to simplify complex class diagrams, you can group classes into packages. A package is a collection of logically related UML elements.

The diagram below is a business model in which the classes are grouped into packages:

- Packages appear as rectangles with small tabs at the top.
- The package name is on the tab or inside the rectangle.
- The dotted arrows are dependencies.
- One package depends on another if changes in the other could possibly force changes in the first.



Basic Concepts of Package Diagram

Package diagram follows hierarchal structure of nested packages. Atomic module for nested package are usually class diagrams. There are few constraints while using package diagrams, they are as follows.

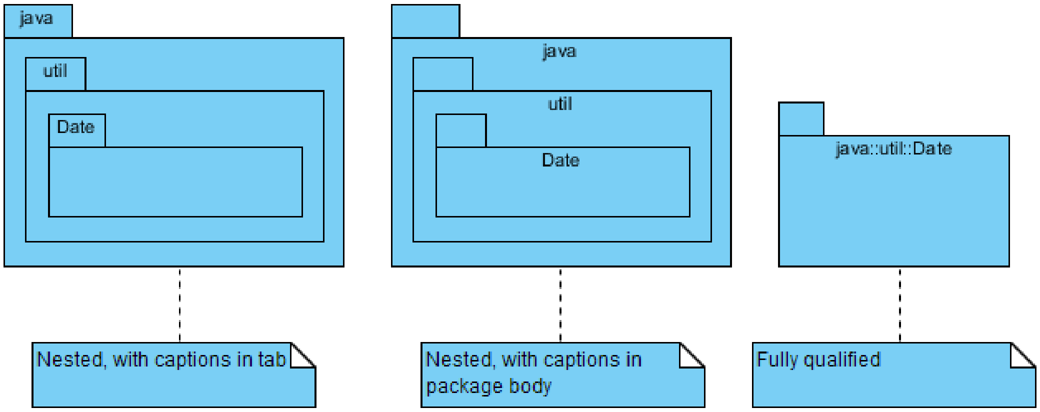
- Package name should not be the same for a system, however classes inside different packages could have the same name.

- Packages can include whole diagrams, name of components alone or no components at all.
- Fully qualified name of a package has the following syntax.

Name owing the package : : Name of the package

java : : util : : Date

Packages can be represented by the notations with some examples shown below:

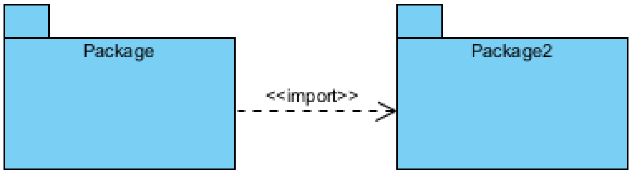

java
util
Date
java
util
Date
java::util::Date
Nested, with captions in tab
Nested, with captions in package body
Fully qualified

Package Diagram - Dependency Notation

There are two sub-types involved in dependency. They are <<import>> & <<access>>. Though there are two stereotypes users can use their own stereotype to represent the type of dependency between two packages.

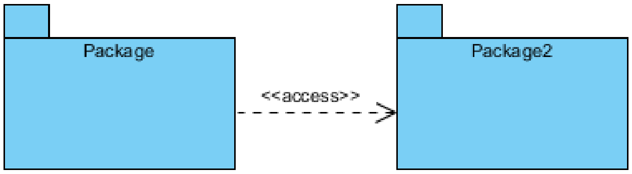
Package Diagram Example - Import

<<import>> - one package imports the functionality of other package



Package Diagram Example - Access

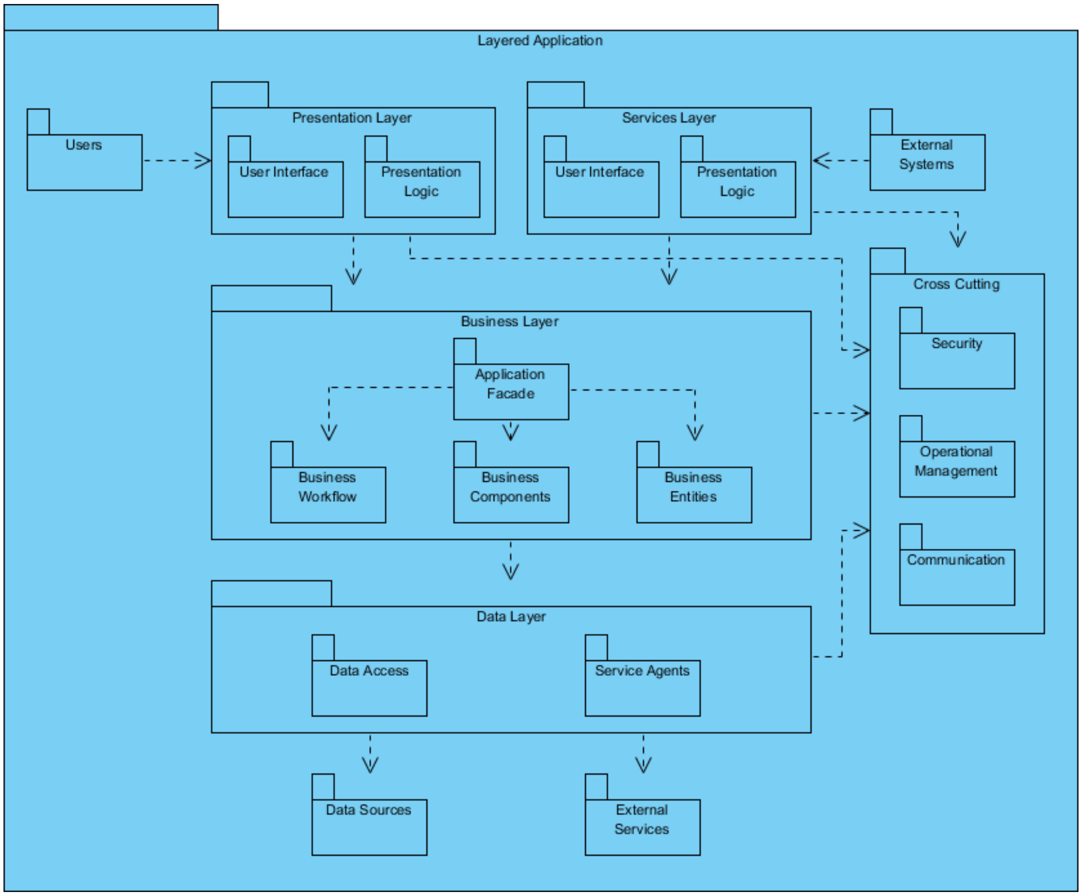
<<access>> - one package requires help from functions of other package.



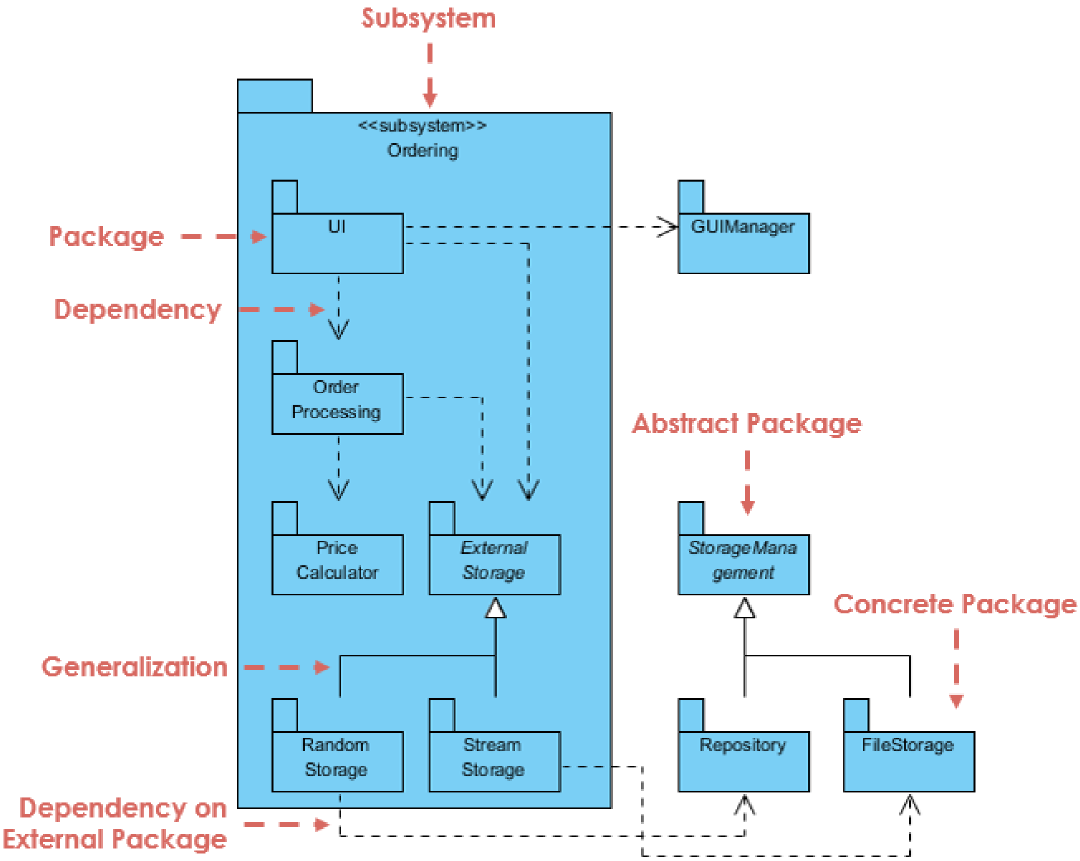
Modeling Complex Grouping

A package diagram is often used to describe the hierarchical relationships (groupings) between packages and other packages or objects. A package represents a namespace.

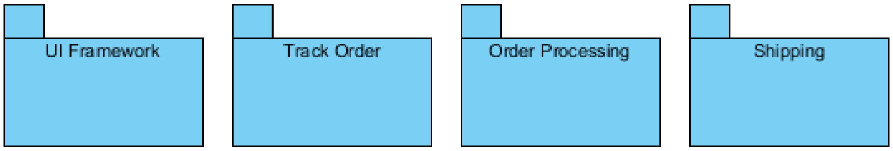
Package Diagram Example - Layering Structure



Package Diagram Example - Order Subsystem



Package Diagram Example - Order Processing System



Package Diagram Example - Order Processing System

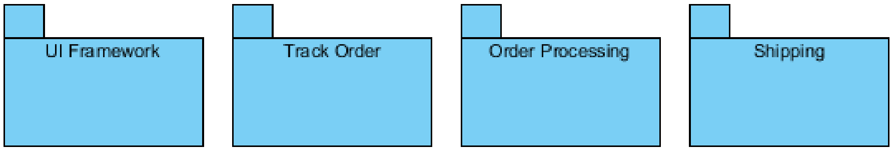
Order Processing System - The Problem Description

We are going to design package diagram for "Track Order" scenario for an online shopping store. Track Order module is responsible for providing tracking information for the products ordered by customers. Customer types in the tracking serial number, Track Order modules refers the system and updates the current shipping status to the customer.

Based on the project Description we should first identify the packages in the system and then related them together according to the relationship:

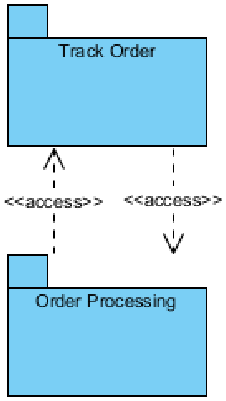
Identify the packages of the system

- There is a track order module, it has to talk with other module to know about the order details, let us call it "Order Details".
- Next after fetching Order Details it has to know about shipping details, let us call that as "Shipping".

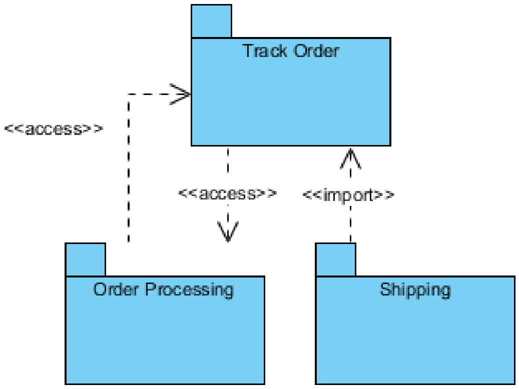


Identify the dependencies in the System

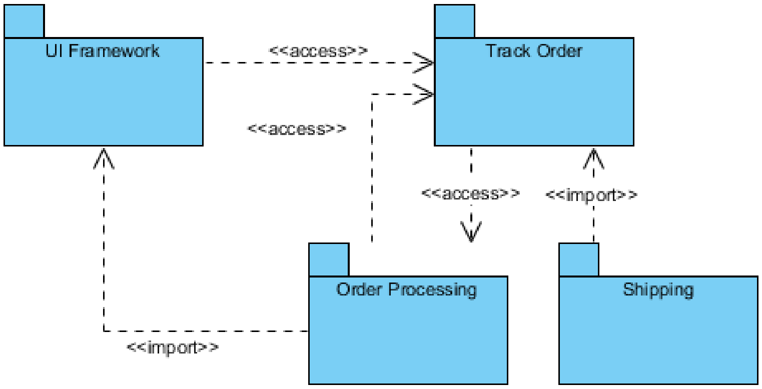
- Track order should get order details from "Order Details" and "Order Details" has to know the tracking info given by the customer. Two modules are accessing each other which suffices <<access>> dual dependency



- To know shipping information, "Shipping" can import "Track Order" to make the navigation easier.



- Finally, Track Order dependency to UI Framework is also mapped which completes our Package Diagram for Order Processing subsystem.



Try to Draw UML Package Diagram Now

Free Download (/download/community.jsp)

Related Links

1. [What is Unified Modeling Language? (/guide/uml-unified-modeling-language/what-is-uml/)](/guide/uml-unified-modeling-language/what-is-uml/)
2. [Professional UML diagram tool (/features/uml-tool/)](/features/uml-tool/)

Turn every software project into a successful one.

Try Visual Paradigm Free (/download/)

(/)

Product

Features (/features/)

Editions (/editions/)

Try Now (/download/)

Pricing (/shop/)

Visual Paradigm Online (https://online.visual-paradigm.com/)

Support

Forums (http://forums.visual-paradigm.com/)

Request Help (/support/#support-form)

Customer Service (https://cs.visual-paradigm.com/)

Learn

Community Circle (https://circle.visual-paradigm.com/)

Know-how (https://knowhow.visual-paradigm.com/)

Demo Videos (/features/demo/)

Tutorials (/tutorials/)

Documents (/support/documents/)

About Us

Visual Paradigm (/aboutus/)

Newsroom (/aboutus/newsreleases/)

YouTube Channel (https://www.youtube.com/us

Academic Partnership (/partner/academic/)

Legal (/aboutus/legal.jsp)

Privacy statement (/aboutus/privacy.jsp)

(https://twitter.com/visualparadigm

(https://www.facebook.com/Visual-Paradigm-822068561487170/)

(https://www.linkedin.com/company
paradigm)

(https://www.pinterest.com/visualpa

(https://www.youtube.com/user/Visu

(https://www.instagram.com/visualp