Getting Started with Eiffel

Eiffel resources

- Follow **resources** link from www.eecs.yorku.ca/course/3311
  Then follow the appropriate link
  - » **Getting eStudio (GPL) for work at home**
  - » **Introduction to programming in Eiffel**
  - » **Input & output**
  - » **Eiffel@York**
    - > **Links to a body of information about Eiffel**
      - – **For those wanting to explore more deeply into Eiffel**

System Components

- Eiffel programs are usually written using **estudio**
  - » **Can also be written using your favourite editor**
    - > **vi, emacs, Xcode, etc.**

- Each class goes in a separate file with extension **.e**

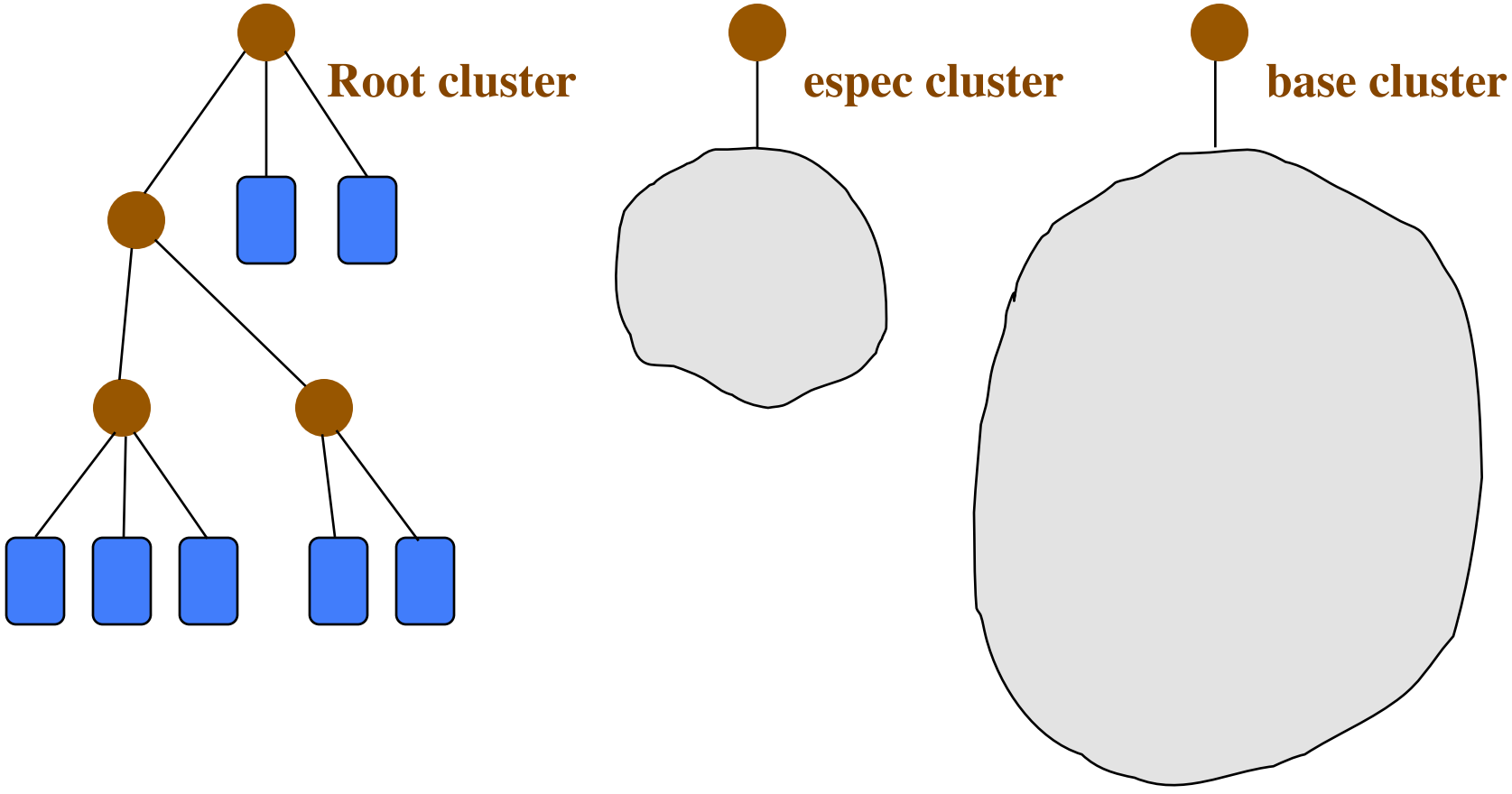
  class_name.e

System Components – 2

- Classes are grouped in clusters
  - » **A clusters is a collection of classes with a unified purpose**
    - > **Input processing**
    - > **Banking**
  - » **Clusters are represented by directories**

- An **ecf** file that specifies the component files for the system
  - » **A simplified make file in XML style**

Directory Structure

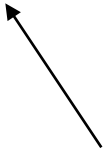


ecf File – Purpose

- To compile and execute a program you need to provide **estudio** with the following information
  - » **The name of the root class and the feature in that class from which execution will begin**
  - » **Identify the set of files and directories that contain classes used by system**
  - » **Specify various system attributes pertaining to assertion checking and other system properties**
  - » **Examples available from the case_studies and pattern_studies links in the sidebar on the course home web page**

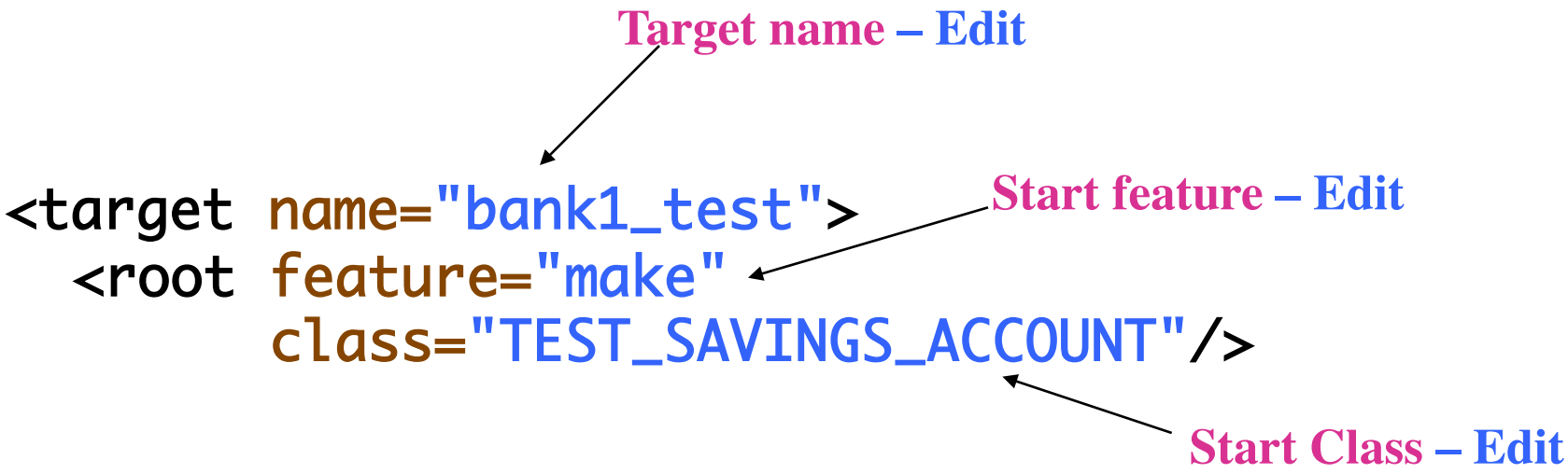
Ecf File – Contents 1 of 3

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<system xmlns=http://www.eiffel.com/developers/xml/configuration-1-9-0
   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
   xsi:schemaLocation="http://www.eiffel.com/developers/xml/
   configuration-1-9-0
   http://www.eiffel.com/developers/xml/configuration-1-9-0.xsd"
   uuid="D1659B65-26A9-4E5B-BDB4-A9C5FF2E8707"

   name="bank1"   >
```



System name
edit for your system

Ecf File – Contents 2 of 3




```
<target name="bank1_test">
  <root feature="make"
        class="TEST_SAVINGS_ACCOUNT"/>
```

```
<option warning="true"  cat_call_detection="false">
   <assertions precondition="true" postcondition="true"
   check="true" invariant="true" loop="true"
   supplier_precondition="true"/>
   </option>
```

Ecf File – Contents 3 of 3

```
<precompile name="base_pre" location="$ESPEC_PRECOMP/base.ecf"/>
</precompile>

<library name="base" location="$ISE_EIFFEL/library/base/
   base.ecf"/>
  <option> <assertions precondition="true"/> </option>
</library>

<library name="espec" location="$ISE_LIBRARY\contrib\library
   \testing\framework\espec_simple\library\espec.ecf"/>


  <cluster name="test" location="./tests"/>
  <cluster name="bank" location="./bank"/>

</target>

</system>
```

All one line

Location of .e files – Edit

ecf File – Creation

- Copy an ecf file, then edit
  - » **To change cluster names and locations**
  - » **Add and delete clusters**
  - » **Change the root class and starting feature**

- Can also use **estudio** to create an **ecf** file by selecting "**Create a new project**" when you startup estudio.
  - » **Then use menu options to add libraries, create clusters and create files**

Eiffel on Prism

- The Eiffel environment and tools on Prism
  - » **/eecs/local/packages/Eiffel14.05**

- Invoke with **estudio14.05**
  - » **Interactive editor – can use others such as emacs**
  - » **Compile and edit options**
  - » **Documentation links on course resources page**
  - » **Familiarize yourself with estudio – it is a powerful system**

Notes

- DO NOT use
  - » **estudio &**
    - > **sysin and sysout will not work with estudio in the background**
- Each instance of estudio can only work with one system (project) at a time.
  - » **To run two or more systems simultaneously requires starting an instance of estudio for each system.**
- Can run estudio from any location but since it can only run one system, it is best to
  - » **Have the ecf file in the root cluster**
  - » **Start estudio from the root cluster for the system**

Eiffel at Home

- Getting Eiffel for a personal computer
  - » **Free ISE Eiffel – sufficient for the course – can be downloaded**
    - > **See the link in the resources web page for the course**

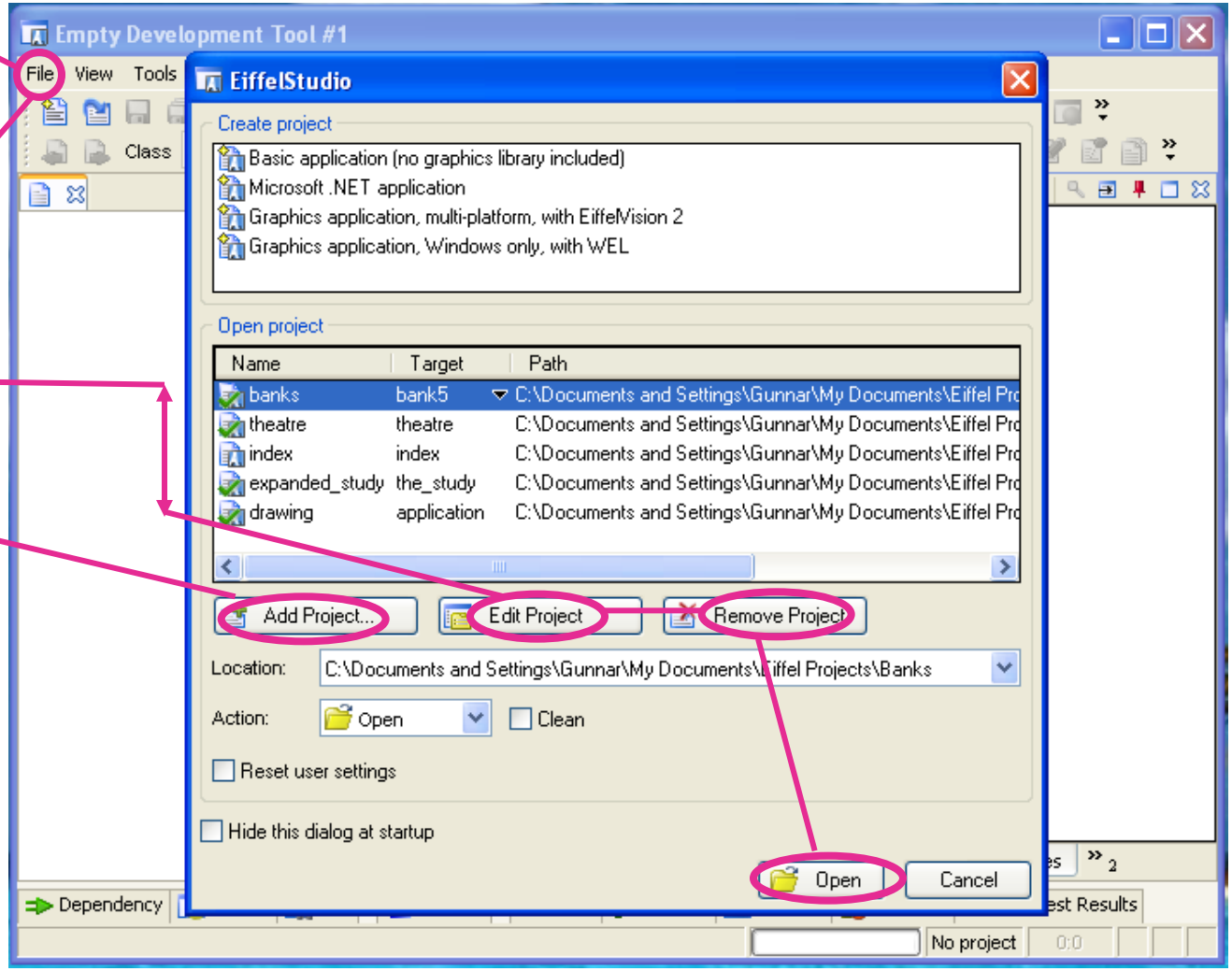
Initial estudio window

Select new project if no ecf file exists

Select open project if ecf file exists

Select a project to open, edit or remove

Select add to browse for an ecf file



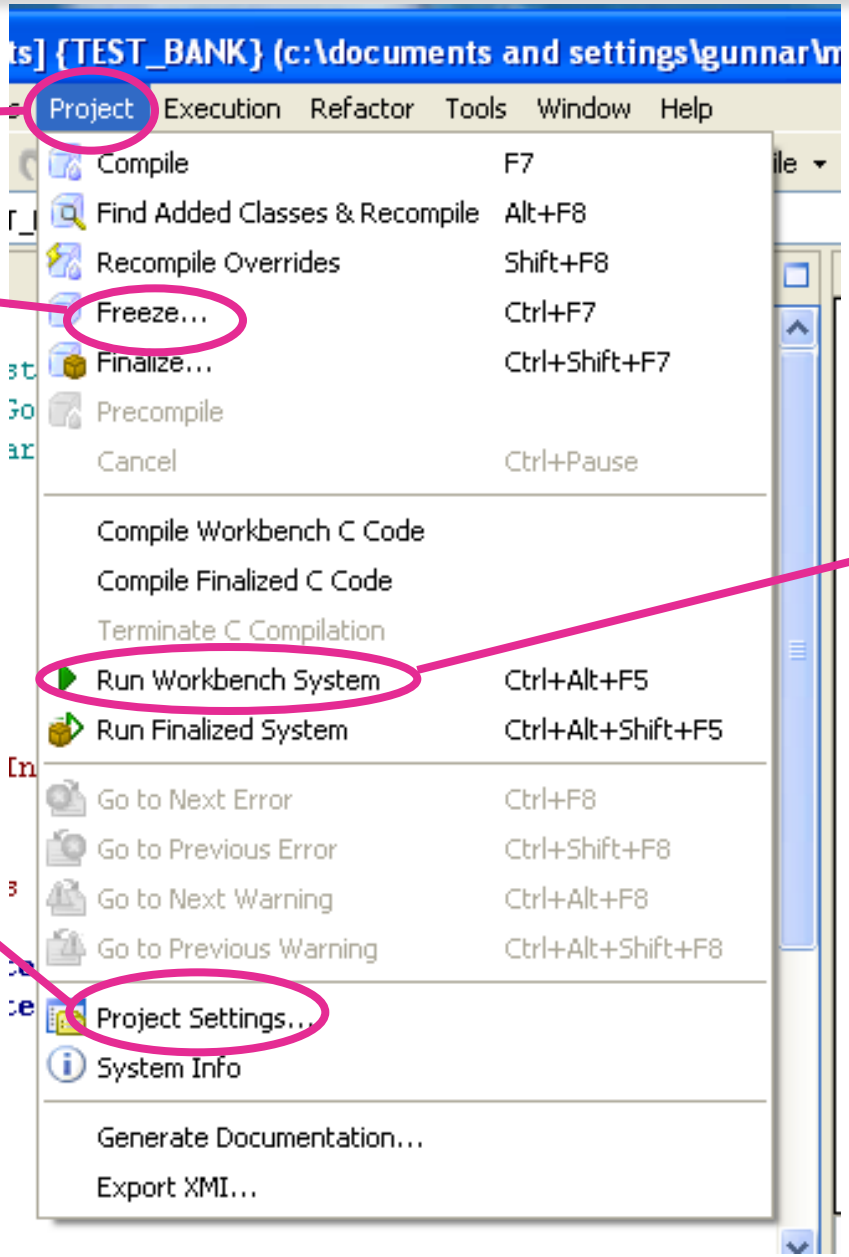
Project Tools

Under

Use if espec terminates with an error

Modify as needed
Edits the ecf file

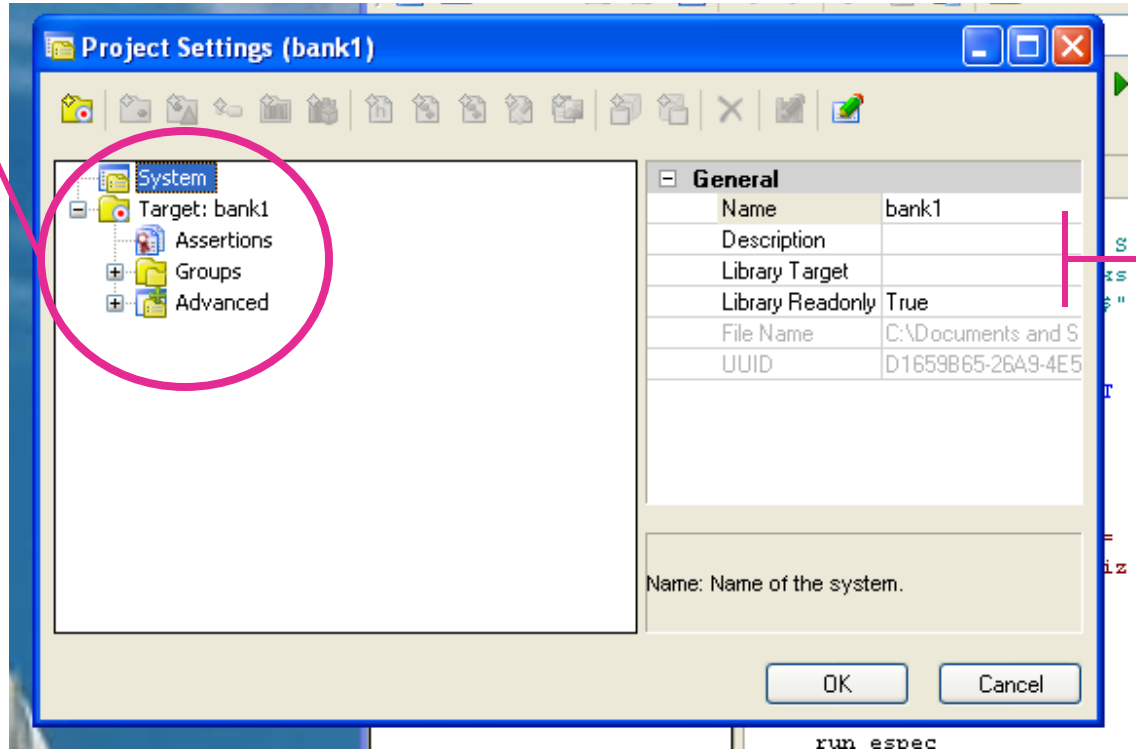
Run eSpec tests



Modify project settings

Select item for
a setting category

Edit fields for your system

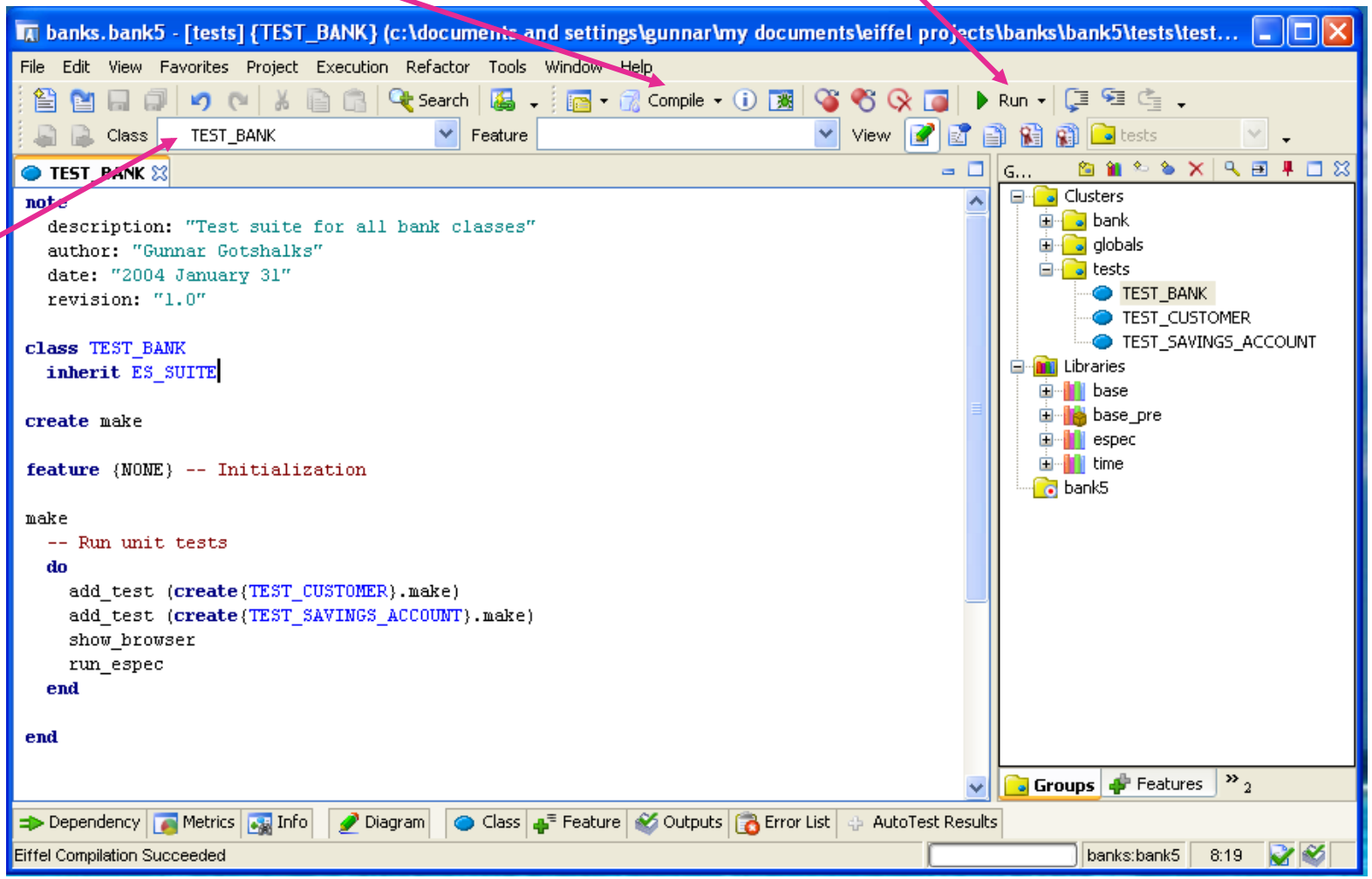


Useful buttons

Compile your program

Run the program, stop at breakpoints

Create a new class by typing a new name

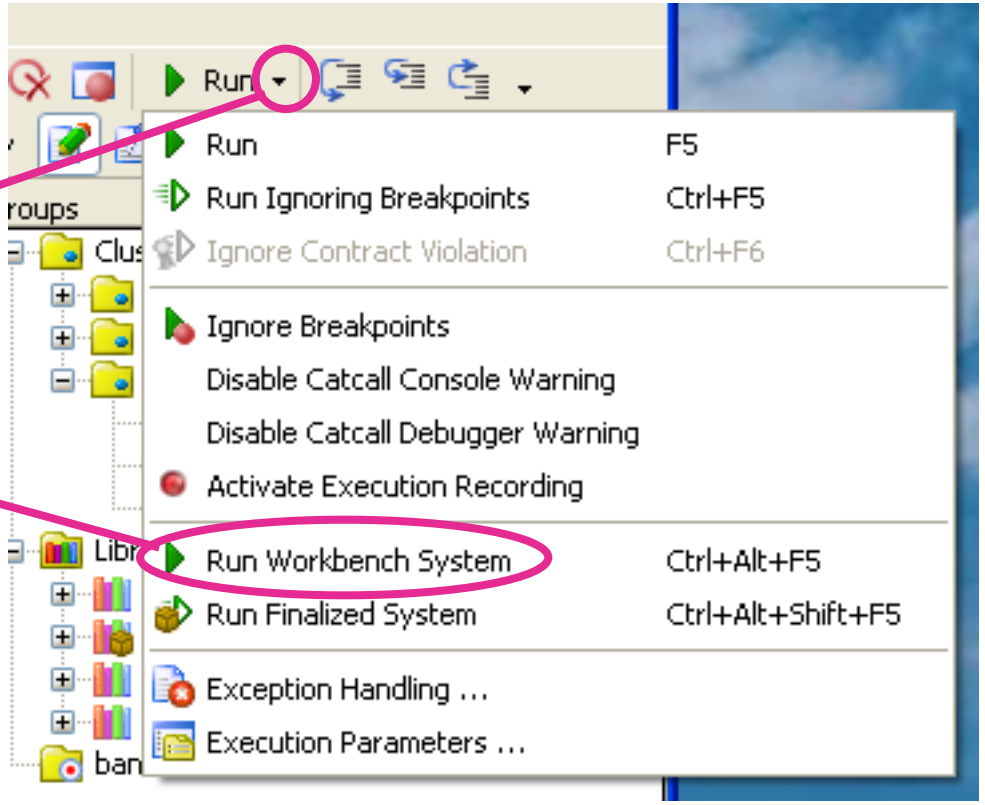


Useful buttons – 2

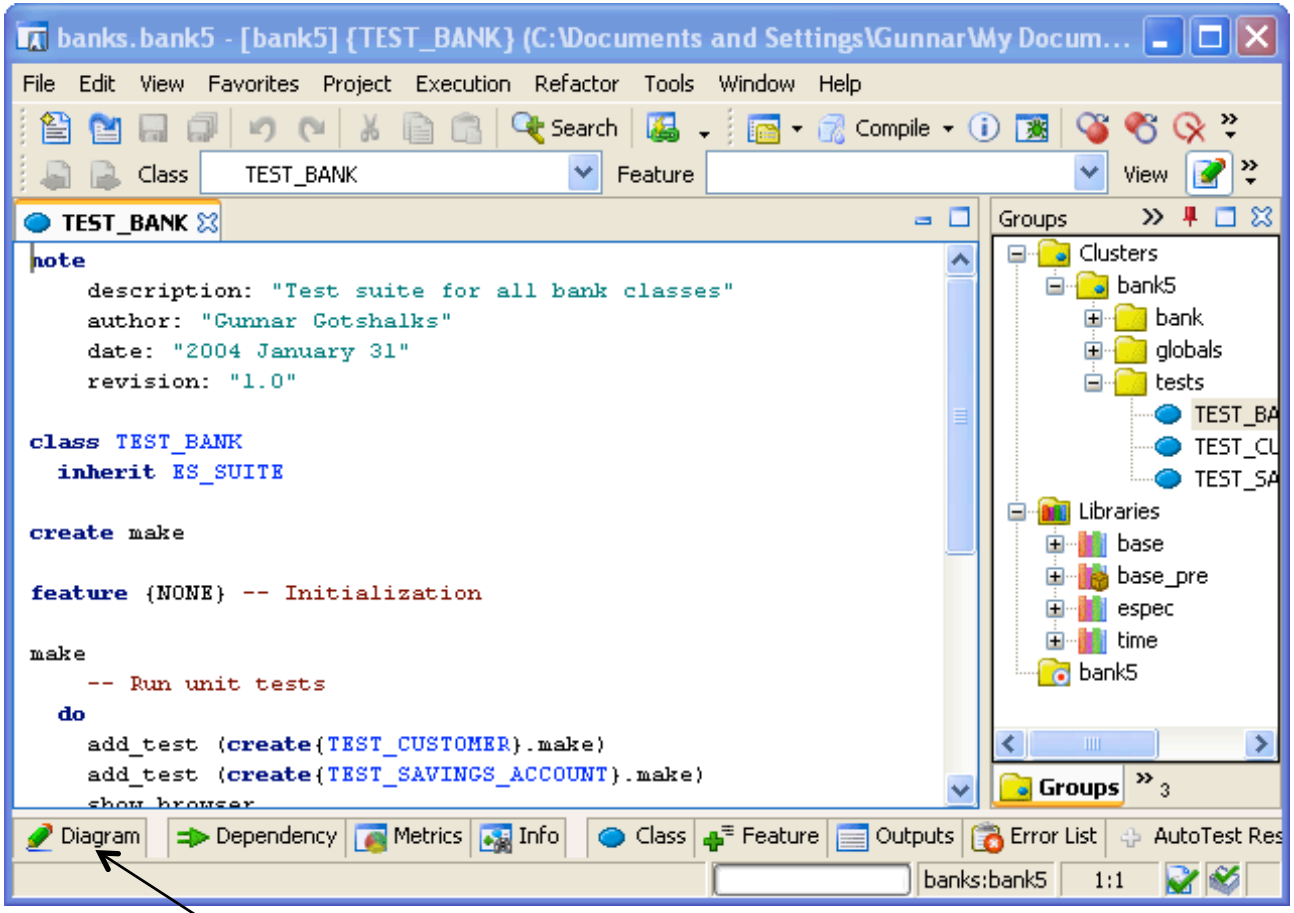
Run eSpec tests

Select triangle

Select

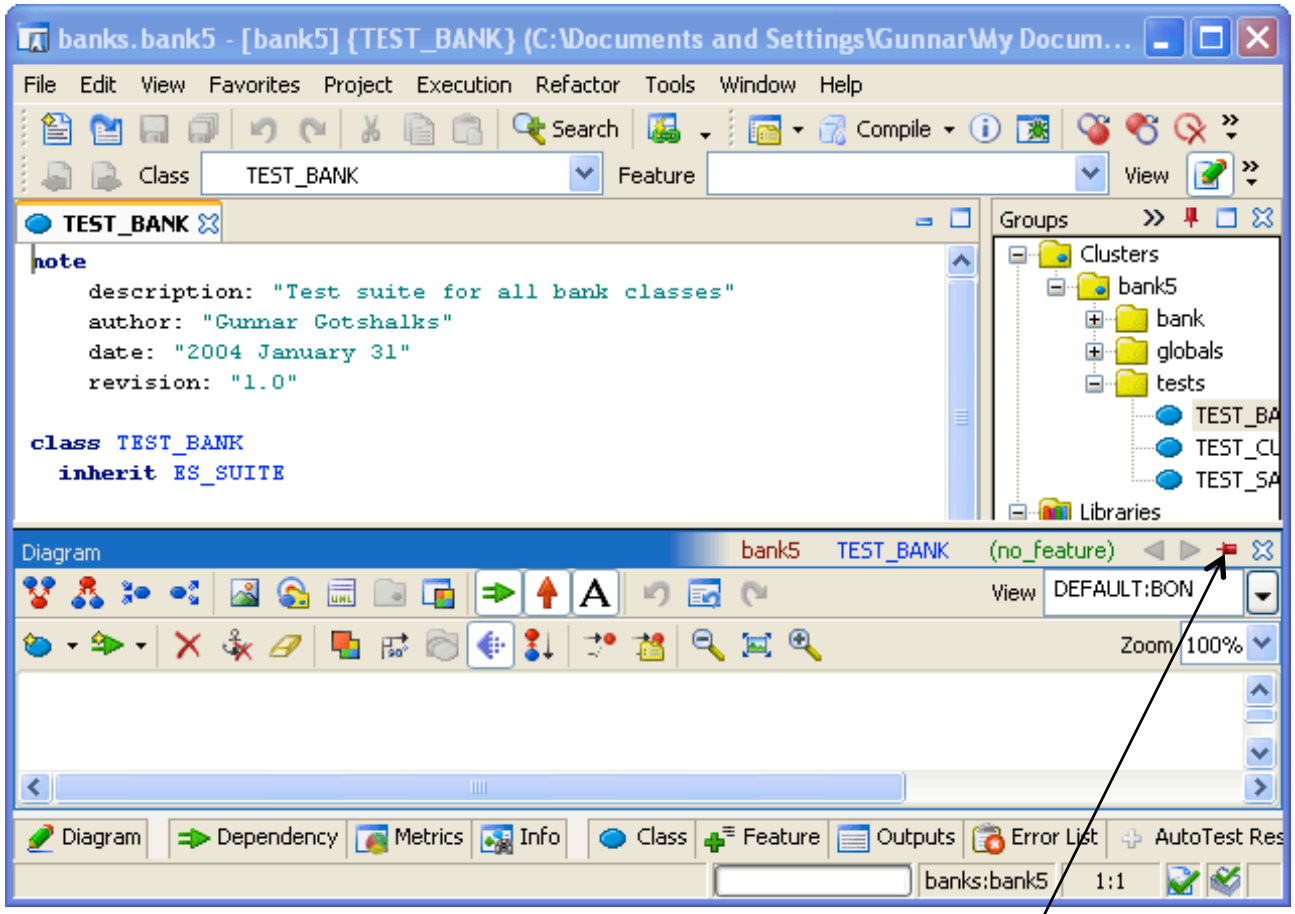


Drawing BON Static Diagrams



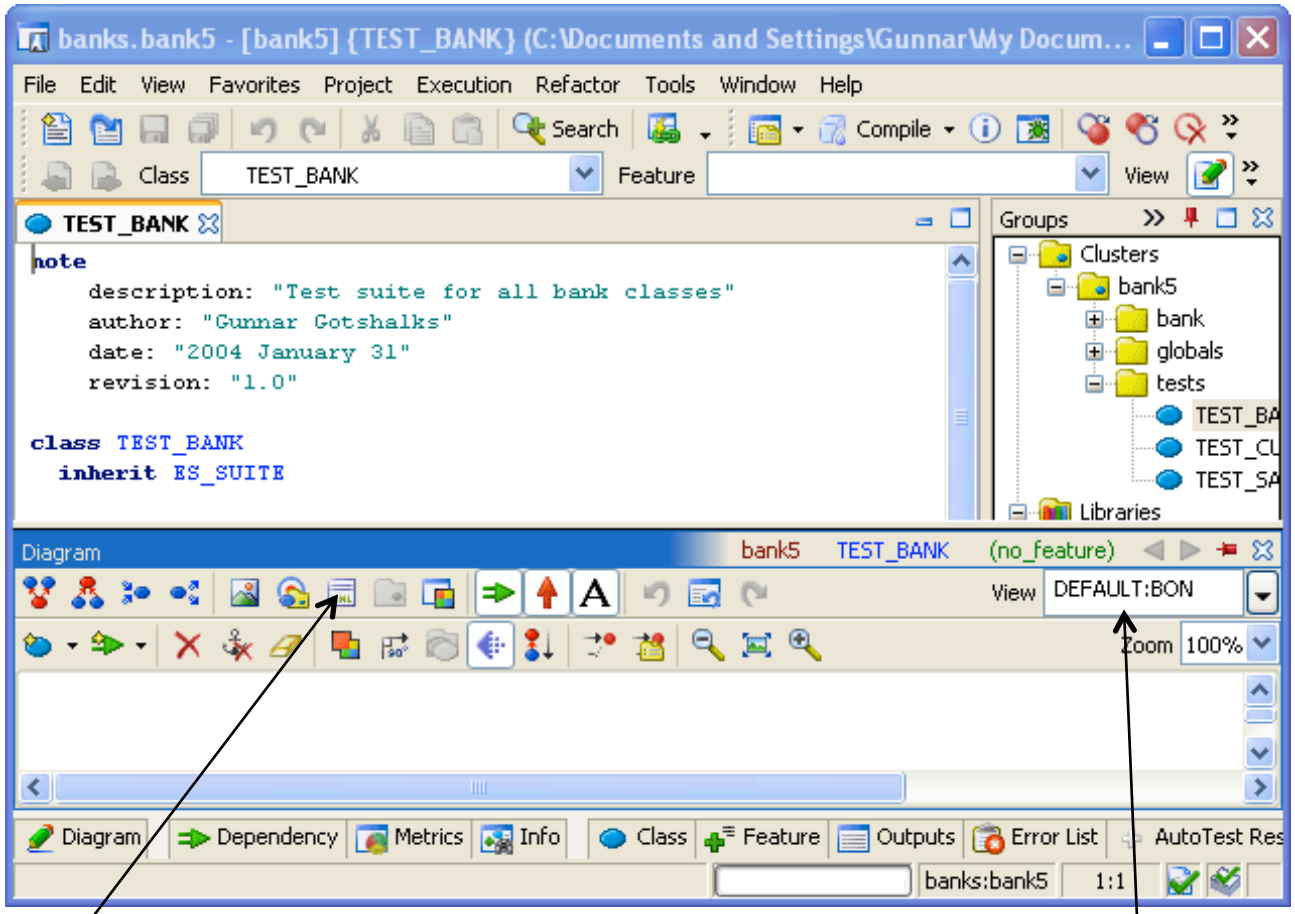
Click on Diagram – could be anywhere among the tabs

Drawing BON Static Diagrams – 2



Pin the diagram

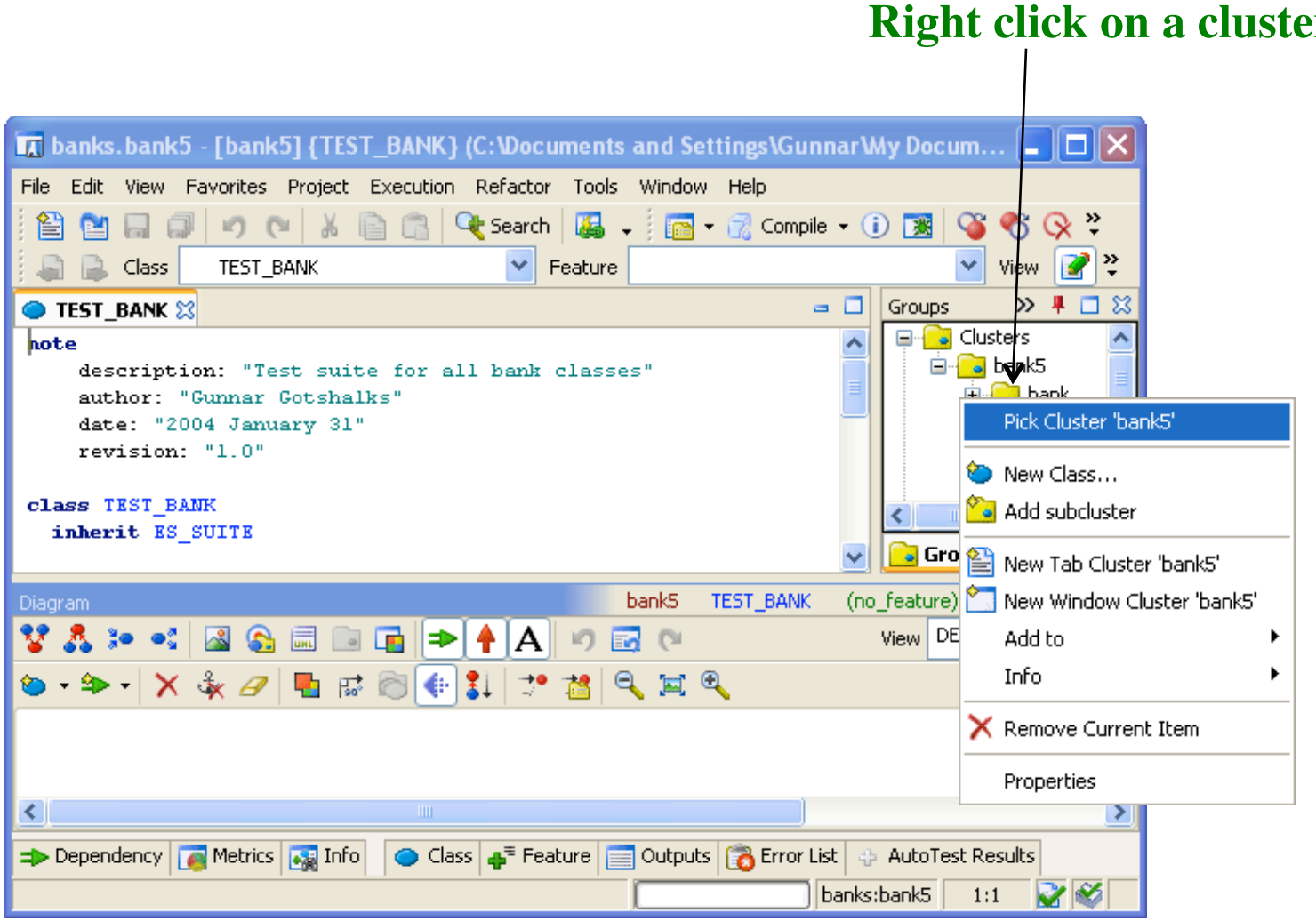
Drawing BON Static Diagrams – 3



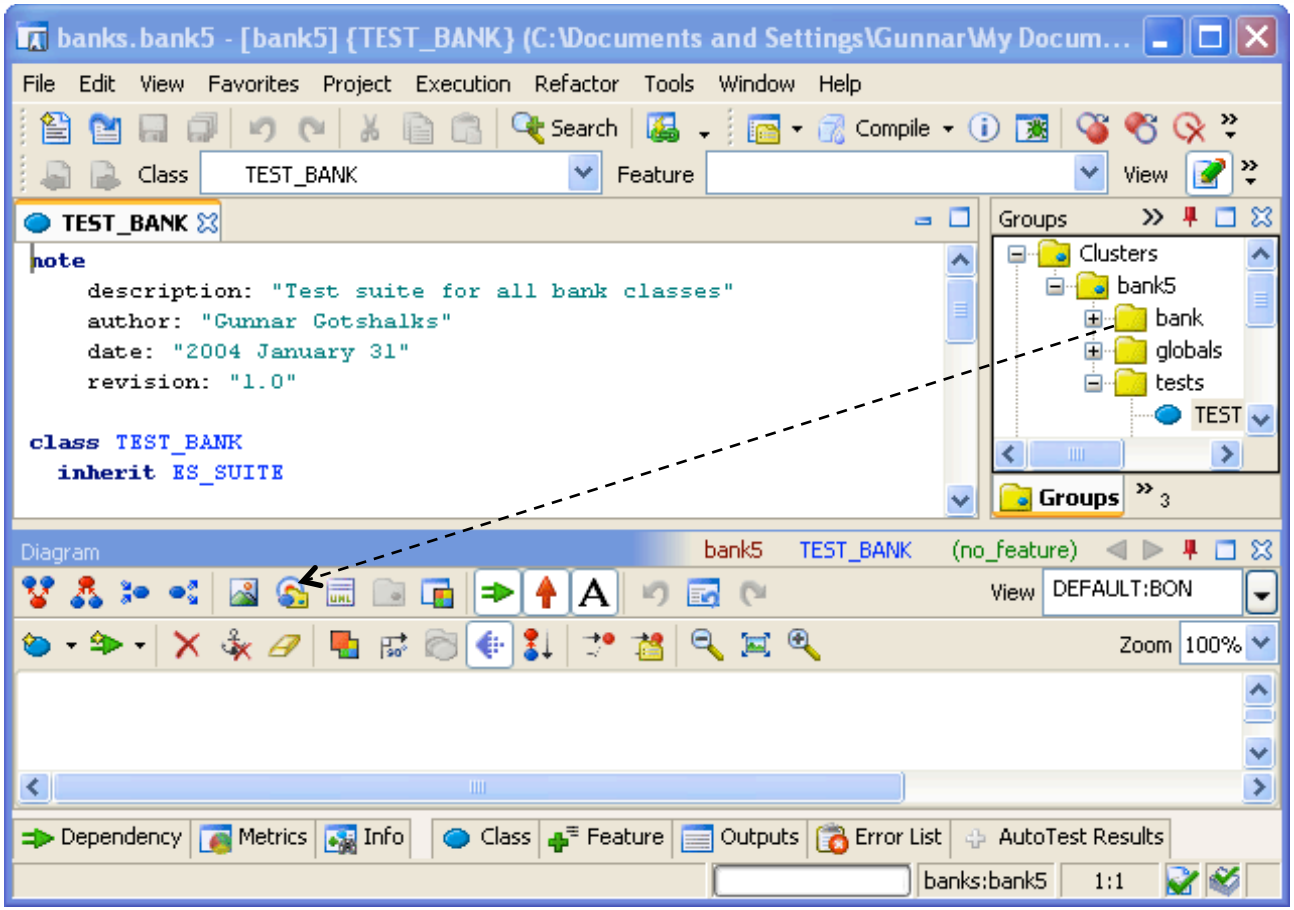
Click this until is says "show UML

This shows DEFAULT: BON

Drawing BON Static Diagrams – 4

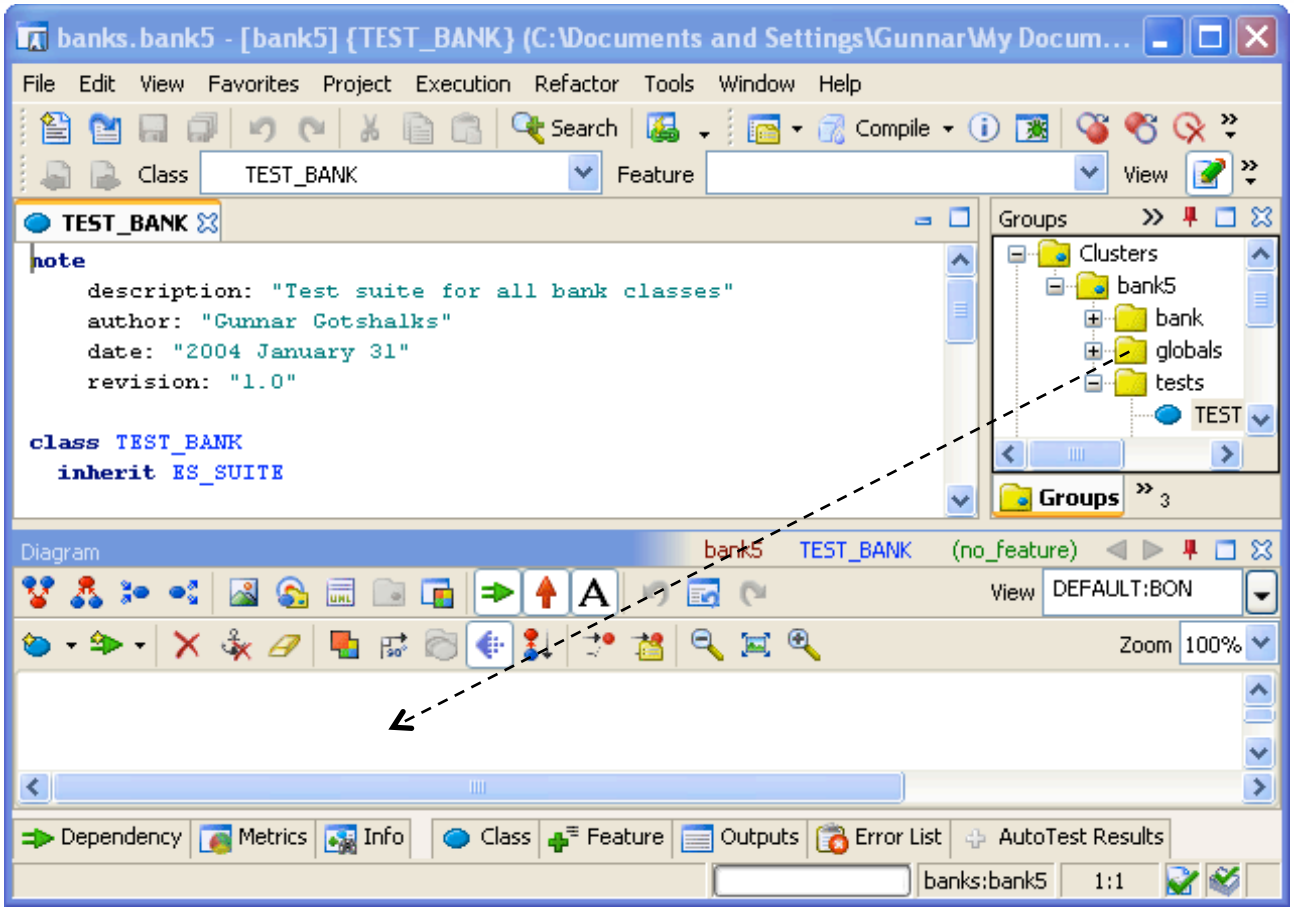


Drawing BON Static Diagrams – 5



Move mouse so arrow of rubber band is on icon at the tip of the arrow (it will be the only one highlighted). Right click again

Drawing BON Static Diagrams – 6



To add another cluster "Pick it" and right click in the Diagram area.