Getting Started with Eiffel

Eiffel resources

- Follow **resources** link from www.cs.yorku.ca/course/3311
  Then follow the appropriate link

  - » **Introduction to programming in Eiffel**

  - » **Eiffel material common to 3311 courses**

    - > **links for downloading Eiffel and other course related material**

  - » **Input & output**

  - » **Brief intro to using estudio**

  - » **Object-oriented programming in Eiffel**

  - » **Eiffel@york**

    - > **links to a large body of information about Eiffel – only 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, etc.)**

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

  class_name.e

- Classes are grouped in clusters

  » **A clusters is a collection of classes with a unified purpose**

  > **Input processing, conference registration**

  » **Clusters are represented by directories**

- An **ecf** file that specifies the component files for the system

  » **A simplified kind of make file in XML style**

Directory Structure

Root cluster

espec cluster

base cluster

⬤ **directory –> cluster**   🟦 **file –> class.e**

03-4
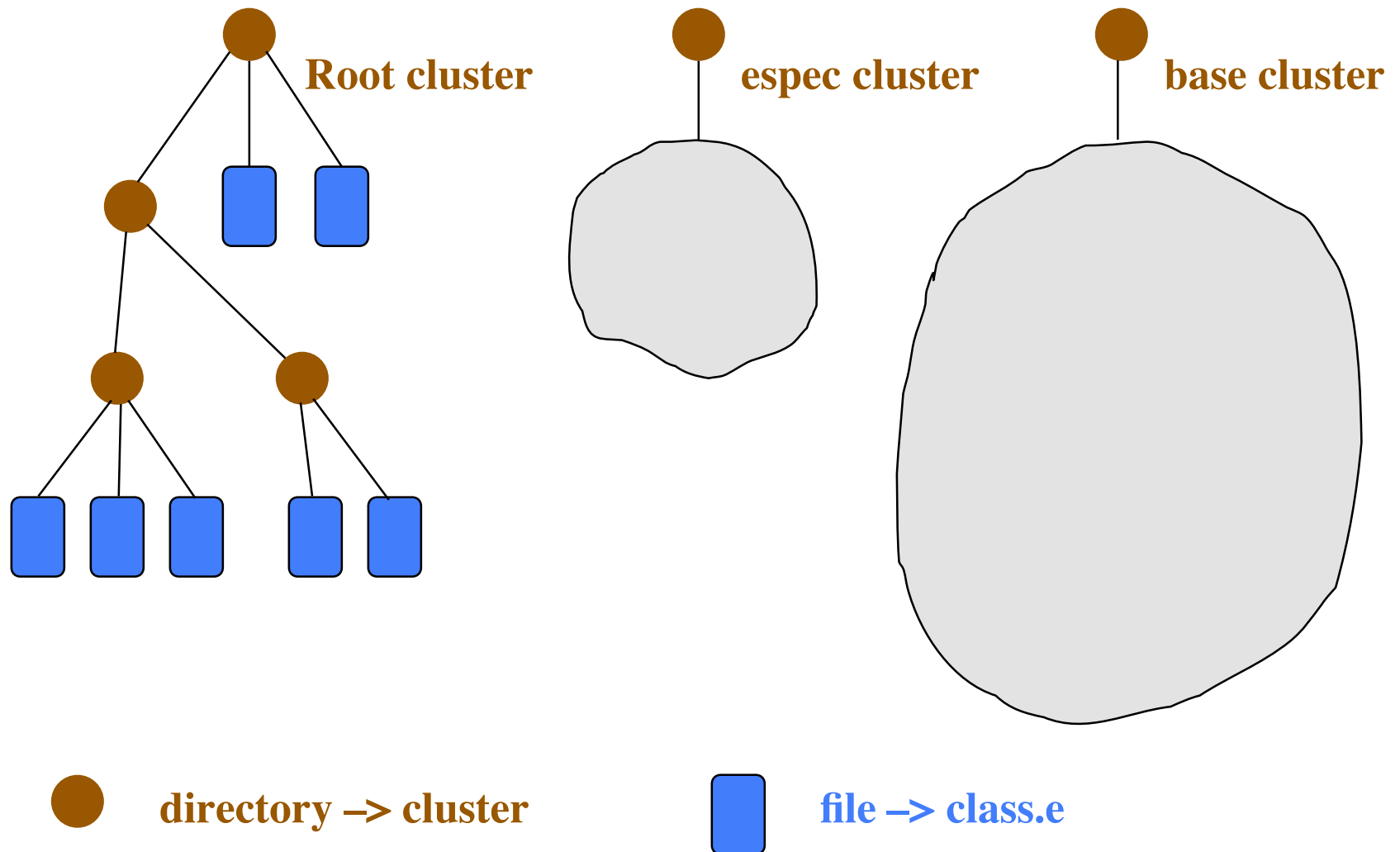
ecf File – Purpose
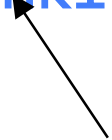
- To compile and execute a program you need to provide **estudio** with the following information

  » **the name of the root class and which 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**

Ecf File – Contents 1 of 3

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

`<system` xmlns=http://www.eiffel.com/developers/xml/configuration-1-0-0
   xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
   xsi:schemaLocation="http://www.eiffel.com/developers/xml/configuration-1-0-0
   http://www.eiffel.com/developers/xml/configuration-1-0-0.xsd"

   name="bank1"   uuid="D1659B65-26A9-4E5B-BDB4-A9C5FF2E8707" `>`

System name
edit for your system

Ecf File – Contents 2 of 3

Target name – Edit

Start feature – Edit

Start Class – Edit

```
<target name="bank1">
  <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>
```

```
<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="$ESPEC/library/espec.ecf"/>


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

</target></system>
```

Location of .e files – Edit

© Gunnar Gotshalks

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 (V6.3) and tools on Prism
  - » **/cs/local/packages/Eiffel63**

- Invoke with **estudio63**
  - » **has 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**

  On Windows you can use 6.5 as that is the standard download

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 on Prism – 2

- Use ecf files to describe

  » **The files comprising your system**

  » **How to execute it**

  » **Examples available from the case_studies link in the sidebar on the course web page**

- 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

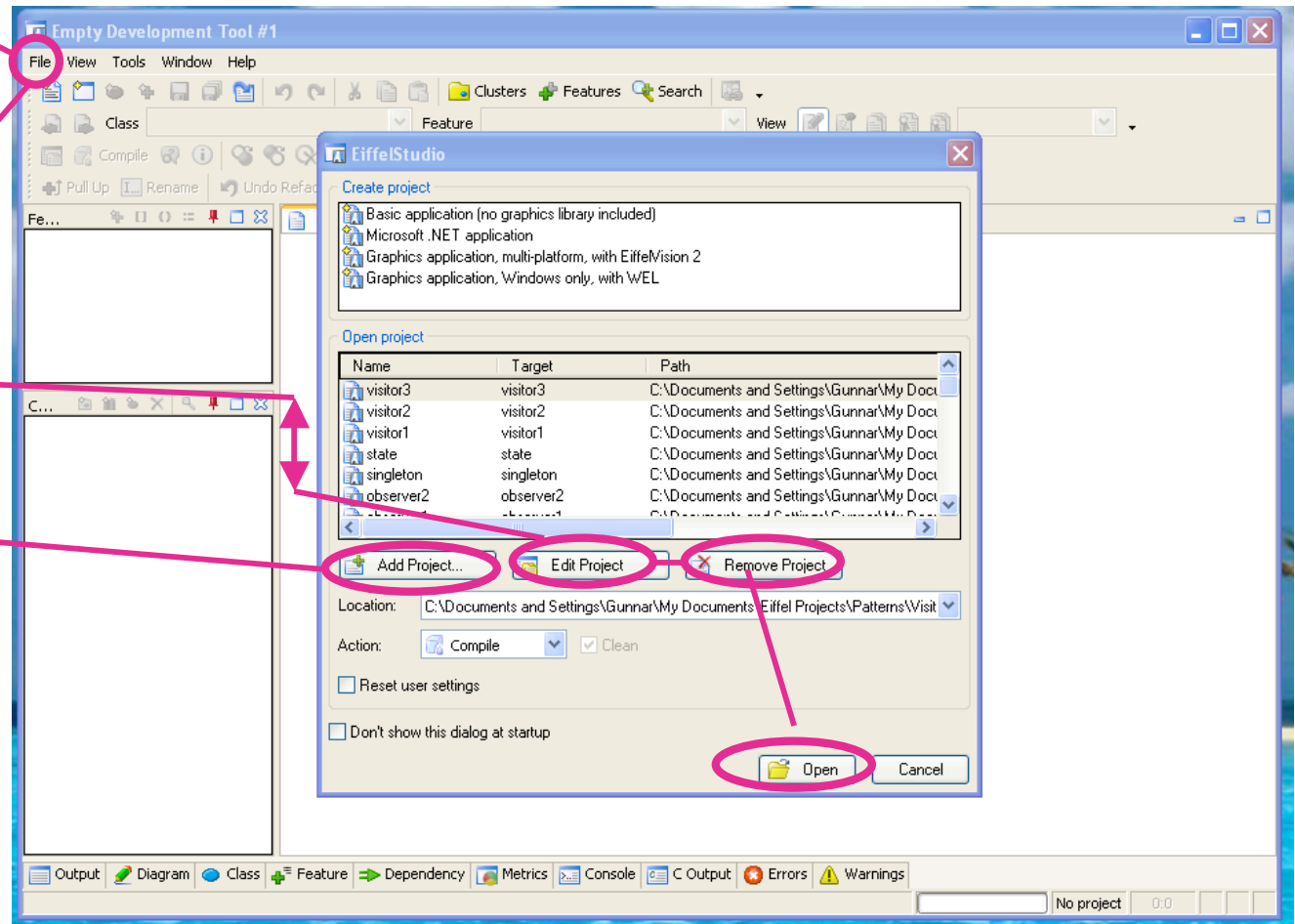 » **SmallEiffel**

  » **Visual Eiffel**

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
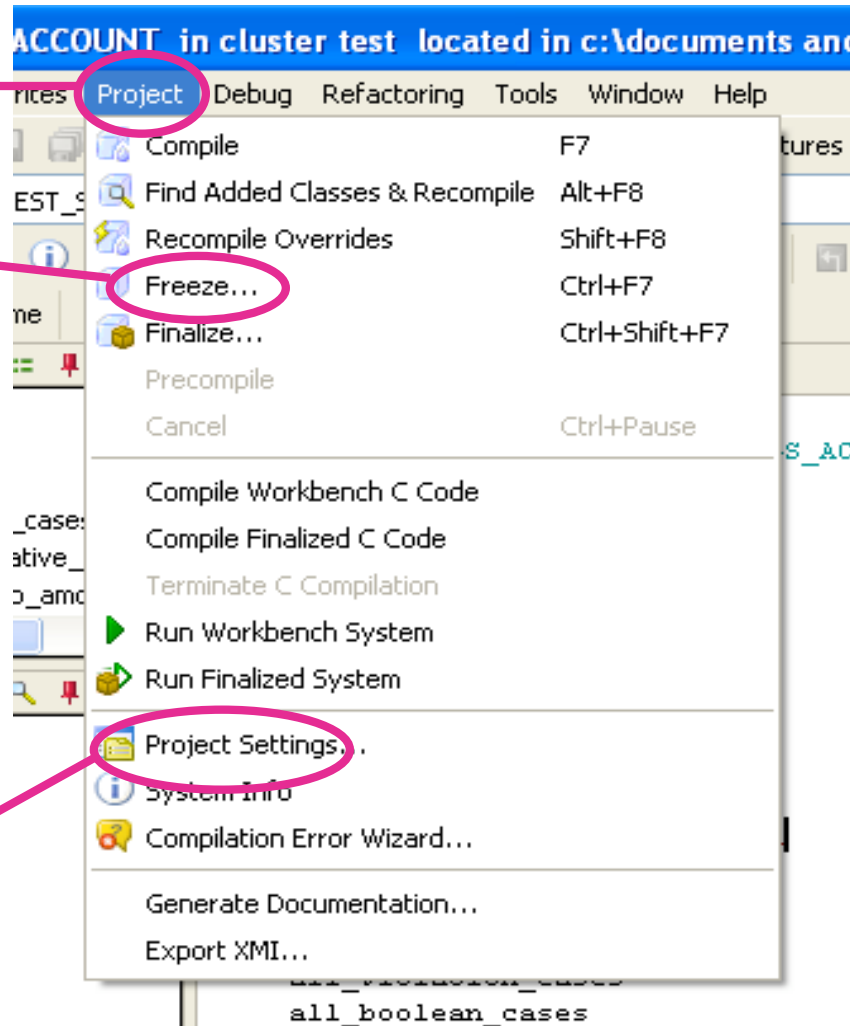
Select add to browse
for an ecf file

Modify project settings
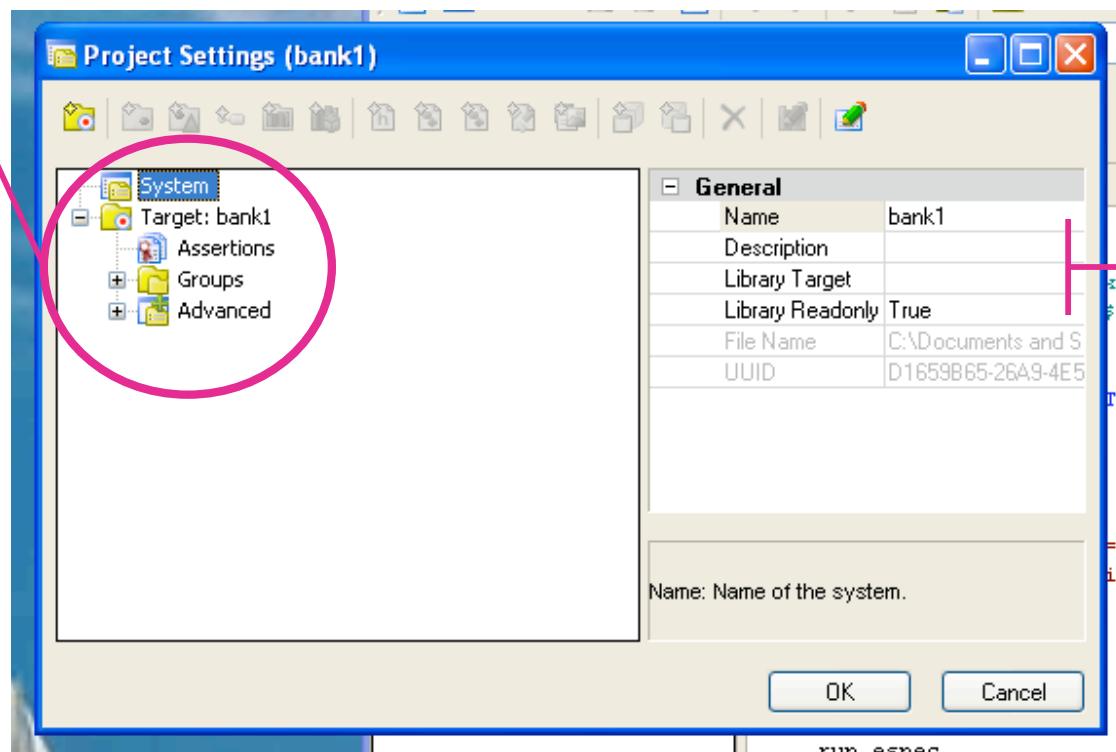
Under

Use if espec terminates with an error

Modify as needed Edits the ecf file

ACCOUNT in cluster test located in c:\documents and

| Project | Debug | Refactoring | Tools | Window | Help |

Compile		F7
Find Added Classes & Recompile		Alt+F8
Recompile Overrides		Shift+F8
Freeze...		Ctrl+F7
Finalize...		Ctrl+Shift+F7
Precompile		
Cancel		Ctrl+Pause

Compile Workbench C Code
Compile Finalized C Code
Terminate C Compilation
Run Workbench System
Run Finalized System

Project Settings...
System Info
Compilation Error Wizard...

Generate Documentation...
Export XMI...

all_boolean_cases

Modify project settings

**Select folder 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**