

## Exercise 7: SCOOP tools.

Hand-out: 7 June  
Due: 14 June

The goal of the exercise is to get acquainted with the SCOOP tools. The **scoop2scoopli** pre-processor and the **SCOOPLI** library will be presented. We will see how a SCOOP project is created, compiled, run, and debugged. We will use a simple producer-consumer example.

### 1. Downloading and installation of tools.

To create, compile, and run your SCOOP projects you will need to download and install the following tools:

- EiffelStudio 5.5
- scoop2scoopli 2.x
- SCOOPLI library 3.x

All downloads and installation instructions can be found on the course web page.

You need to download scoop2scoopli from

<http://se.inf.ethz.ch/teaching/ss2005/0268/project/download/scoop2scoopli.zip>.

No installation is necessary. You can use directly the executable scoop2scoopli.exe.

The only requirement is the presence of the SCOOPLI library version 3.x. Make sure that all environment variables that are used in your Ace files (e.g. ISE\_EIFFEL) are set.

The SCOOPLI library can be downloaded from

<http://se.inf.ethz.ch/teaching/ss2005/0268/project/download/scoopli.zip>.

Extract it to a chosen directory (e.g. C:/eiffel\_libraries/scoopli). Set the environment variable \$SCOOPLI to point to that directory.

### 2. scoop2scoopli

**scoop2scoopli** is a preprocessor and a type checker for SCOOP programs. It translates SCOOP programs into standard Eiffel. The generated system can be compiled with the ISE Eiffel compiler 5.5 or newer. **scoop2scoopli** is a command-line tool.

**scoop2scoopli** performs lexical, syntactical, and semantic analysis of the source code. In fact, it performs a full compilation of your SCOOP system except for the generation of final code in C or IL. Instead, it generates standard Eiffel code that needs to be compiled with ISE Eiffel

compiler. The latter is only used as a back-end, i.e. it should not report any compilation errors.

## HOW TO USE IT?

**Step 1.** To launch the preprocessing of a SCOOP system, type "`scoop2scoopli filename.ace`", where "`filename.ace`" is the Ace file of your SCOOP system.

**scoop2scoopli** analyses all classes in the clusters specified in the Ace file and generates the following directories and files:

- directory “scoop\_build” that is the root directory of the generated project. All subsequently generated directories and files are stored in that directory (or its subdirectories).
- a new Ace file: "`system_name.ace`", where "`system_name`" corresponds to the system name as specified in the original Ace file. If the original Ace file targets the .NET platform, so will the new Ace file; otherwise, native Windows will be targeted.
- directory “original\_classes” that contains all classes from your original system, organised in clusters.
- directory “separate\_classes” that contains classes that implement proxies to separate objects, organised in clusters.
- class file: "`original_classes/cluster_name/class_name.e`" - for each class in your original system. Original class is modified if necessary.
- class file: "`separate_classes/cluster_name/scoop_separate__class_name.e`" - for each class in your original system. This class file contains the class that implements a separate proxy.
- the root class: "`scoop_starter.e`".

**Step 2.** Compile the generated system (Ace file "`scoop_build/system_name.ace`") using the ISE Eiffel compiler.

**Step 3.** Your SCOOP application is compiled. You can run it now.

## 3. SCOOPLI library

SCOOPLI is a library implementation of the SCOOP model [Mey97] for Eiffel. The implementation relies on the multithreading library EiffelThread. It runs on native Windows and .NET 1.1.

SCOOP (Simple Concurrent Object-Oriented Programming) is a concurrency model based on the principles of Design by Contract. The model takes advantage of the inherent concurrency implicit in object-oriented programming to provide programmers with a simple extension enabling them to produce parallel applications with little more effort than sequential ones.

The present document focuses on the SCOOPLI library; for more information about the SCOOP model please visit the website <http://se.inf.ethz.ch/research/scoop.html>.

SCOOPLI 3.0 implements most functionalities of SCOOP:

- Declaration of separate entities
- Enclosing routines

- Separate feature calls
- Passing expanded objects as arguments and results of separate calls
- Passing separate references as arguments and results of separate calls
- Passing non-separate references as arguments and results of separate calls
- Wait-by-necessity

SCOOP features that have not been implemented yet:

- Duels

## **HOW TO USE SCOOPLI?**

There are two ways of using SCOOPLI in your Eiffel projects:

- Write your SCOOP application and preprocess the source code with the **scoop2scoopli** tool. The necessary SCOOPLI classes and an Ace file will be automatically generated for you. The generated code can be then compiled with the ISE Eiffel compiler.
- Use SCOOPLI classes directly in your source code. For masochists.