deeco
– Programmer's Manual –

Thomas Bruckner
Institut für Theoretische Physik der Universität Würzburg

December 19, 1997

DISCLAIMER OF WARRANTY

THE PROGRAM “deeco” IS PROVIDED ”AS IS” WITHOUT WARRANTY OF ANY KIND. WE MAKE NO WARRANTIES, EXPRESS OR IMPLIED, THAT THE PROGRAM IS FREE OF ERROR, OR IS CONSISTENT WITH ANY PARTICULAR STANDARD OF MERCHANTABILITY, OR THAT IT WILL MEET YOUR REQUIREMENTS FOR ANY PARTICULAR APPLICATION. IT SHOULD NOT BE RELIED ON FOR SOLVING A PROBLEM WHOSE INCORRECT SOLUTION COULD RESULT IN LOSS OF PROPERTY. IF YOU DO USE THE PROGRAM IN SUCH A MANNER, IT IS AT YOUR OWN RISK. THE AUTHORS AND PUBLISHER DISCLAIM ALL LIABILITY FOR DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM YOUR USE OF THE PROGRAM.

1 Units

1.1 Header files

Commercial:

Syntax: file name (compiler options) : comment

<iostream.h>	(-I/usr/include/CC)	:	Input-Output Control
<fstream.h>	(-I/usr/include/CC)	:	File Management
<strstream.h>	(-I/usr/include/CC)	:	Stringstream Management
<new.h>	(-I/usr/include/CC)	:	For Use of a new-handler
<Objection.h>	(-I/usr/include/SC)	:	Error Handling
<Stopwatch.h>	(-I/usr/include/SC, -l++)	:	Start-Stop Control
<String.h>	(-I/usr/include/SC, -l++)	:	Easy String Handling
<Symbol.h>	(-I/usr/include/SC, -l++)	:	String-Key Handling
<List.h>	(-I/usr/include/SC, -l++)	:	List Handling
<Map.h>	(-I/usr/include/SC, -l++)	:	Associative Arrays
<Path.h>	(-I/usr/include/SC, -l++)	:	UNIX-Path Management
<Graph.h>	(-I/usr/include/SC, -lGA -lGraph -l++)	:	Graph Management
<Graph_alg.h>	(-I/usr/include/SC, -lGA -lGraph -l++)	:	Graph Algorithms
<time.h>	(-I/usr/include)	:	Time and Date Handling
<assert.h>	(-I/usr/include)	:	Control of Assertions
<ctype.h>	(-I/usr/include)	:	Control of Characters
<stdlib.h>	(-I/usr/include)	:	For Use of Exit Function
<stdio.h>	(-I/usr/include)	:	Standard Input-Output Functions
<signal.h>	(-I/usr/include)	:	For Catching the Interrupt Signal
<math.h>	(-I/usr/include)	:	Math Functions

- iostream.h : Input-Output Control
  Library: C++ Stream Library
  /usr/include/CC/iostream.h
  Info: Online manual

- fstream.h : Filemanagement
  Library: C++ Stream Library
  /usr/include/CC/fstream.h
  Info: Online manual

- strstream.h : Stringstream - management
  Library: C++ Stream Library
  /usr/include/CC/strstream.h
  Info: Online manual

- new.h : For Use of a new-handler
  Library: C++ Stream Library
  /usr/include/CC/new.h
  Info:

- Objection.h: Unified error-handling (used by other components of the Standard Components Library to report errors)
  Library: Standard Components Library
  /usr/include/SC/Objection.h
  Info: Online manual: /usr/CC/man/SC

- Stopwatch.h : Start-Stop-Control
  Library: Standard Components Library
  /usr/include/SC/Stopwatch.h
  Info: Online manual: /usr/CC/man/SC

- String.h : Easy String handling
  Library: Standard Components Library
  /usr/include/SC/String.h
  Info: Online manual: /usr/CC/man/SC

- Symbol.h : String-Key handling
  Library: Standard Components Library
  /usr/include/SC/Symbol.h
  Info: Online manual: /usr/CC/man/SC

- List.h : List handling
  Library: Standard Components Library
  /usr/include/SC/List.h
  Info: Online manual: /usr/CC/man/SC

- Map.h : Associative Arrays
  Library: Standard Components Library
  /usr/include/SC/Map.h
  Info: Online manual: /usr/CC/man/SC

- Path.h : UNIX-Path Mangement
  Library: Standard Components Library
  /usr/include/SC/Path.h
  Info: Online manual: /usr/CC/man/SC

- Graph.h : Graph Management
  Library: Standard Components Library
  /usr/include/SC/Graph.h
  Info: Online manual: /usr/CC/man/SC

- Graph_alg.h : Graph Algorithms
  Library: Standard Components Library
  /usr/include/SC/Graph_alg.h
  Info: Online manual: /usr/CC/man/SC especially: online manual for artic_pts, bfs, comps, cycle, cycle_list, dfs.

- time.h : Time and Date handling
  HP-UX C-Library /usr/include/time.h
  Info: Online manual and (lrom) HP-UX Reference - Vol. 2 Section 3

- assert.h : Control of Assertions (simple error handling)
  HP-UX C-Library /usr/include/assert.h
  Info: Online manual and (lrom) HP-UX Reference - Vol. 2 Section 3

- ctype.h : Control of Characters
  HP-UX C-Library /usr/include/ctype.h
  Info: Online manual and (lrom) HP-UX Reference - Vol. 2 Section 3

- stdlib.h : For Use of ”Exit”
  HP-UX C-Library usr/include/stdlib.h
  Info: Online manual and (lrom) HP-UX Reference - Vol. 2 Section 2

- stdio.h : Standard Input-Output Functions
  HP-UX C-Library usr/include/stdio.h
  Info: Online manual and (lrom) HP-UX Reference - Vol. 2 Section 3

- signal.h : Specify What to do upon Receipt of a Signal
  HP-UX C-Library usr/include/signal.h
  Info: Online manual and (lrom) HP-UX Reference - Vol. 2 Section 2

- math.h : C Math Functions
  HP-UX C-Math Library usr/include/math.h
  Info: Online manual and (lrom) HP-UX Floating Point Guide, Appendix A

User definied:

App.h	:	General Application Objects
Balan.h	:	Deeco Energy Balance Objects
Collect.h	:	Deeco Collector-Process Modules
Connect.h	:	Deeco Process Connection Management Objects
Convers.h	:	Deeco Conversion-Process Modules
DGraph.h	:	Deeco User Definied Graph Management Objects
Data.h	:	Deeco Data Management Objects
Demand.h	:	Decco Demand-Process Modules
Net.h	:	Deeco Energy Supply Network Management Objects
Network.h	:	Decco Network-Process Modules
Port.h	:	Deeco Import-Export-Process Modules
Proc.h	:	Deeco Process Management Objects
ProcType.h	:	Deeco Process Type Management Objects
Scen.h	:	Deeco Scenario Management Objects
Simplex.h	:	Simplex Algorithm Management Objects
Storage.h	:	Deeco Storage-Process Modules
deecoApp.h	:	Deeco Application Objects
nr.h, nrutil.h	:	Numerical Recipes Headers

1.2 Source Code Files

App.C	:	General Application Objects
Balan.C	:	Deeco Energy Balance Objects
Collect.C	:	Deeco Collector-Process Modules
Connect.C	:	Deeco Process Connection Management Objects
Convers.C	:	Deeco Conversion-Process Modules
DGraph.C	:	Deeco User Definied Graph Management Objects
Data.C	:	Deeco Data Management Objects
Demand.C	:	Decco Demand-Process Modules
Net.C	:	Deeco Energy Supply Network Management Objects
Network.C	:	Decco Network-Process Modules
Port.C	:	Deeco Import-Export-Process Modules
Proc.C	:	Deeco Process Management Objects
ProcType.C	:	Deeco Process Type Management Objects
Scen.C	:	Deeco Scenario Management Objects
Simplex.C	:	Simplex Algorithm Management Objects
Storage.C	:	Deeco Storage-Process Modules
deeco.C	:	Main Function of Deeco
deecoApp.C	:	Deeco Application Objects
nrutil.c	:	Help Routines for Numerical Recipes Algorithms
simp1.c,simp2.c,simp3.c	:	Numerical Recipes Simplex Algorithms Help Routines

simplx.c	: Numerical Recipes Simplex Algorithm Routine

2 Global Definitions

2.1 Variables

The following variables are not defined in objects, but as program parts, that are accessible globally:
Syntax: Variable Type Variable name : Unit

App dApp	: deeco.C
String sComandArg	: deeco.C
int comSilentFlag	: deeco.C
int comTestFlag	: deeco.C
int comLogFlag	: deeco.C
String comProjectName	: deeco.C
String comProjectPath	: deeco.C

3 Miscellaneous

3.1 Command line arguments

The following command line arguments are defined:

”-s” : Silent Mode; i.e. no header is displayed
”-t” : Test Mode; detailed program execution information
”-l” : Log Mode, write log file
”-d” : Project path (full directory name); should be used after ”-d” without any space)
”-n” : Project name; should be used after ”-n” without any space)

3.2 Log file

Program execution informations are dispayed on the screen and written in a log file called ”deeco.log”.

3.3 Test

If ” #define _TEST_” is used as a compiler option in the file TestFlag.h then extensive information will be displayed. If runtime is crucial the programm ”deeco” should be compiled without this option (delete the line ” #define _TEST_”).

3.4 Error handling

The program deeco is provided with an error/warning/info message file called ”deeco.msg”. The length of all entries in that file should be lower than 254 characters. The function dApp.writeMessage is used to display the error message. If the operation ”new” inside of this function fails, the message ”Assertion failed ...” (see assert.h) will be displayed. For error numbers larger than 1000 information messages and for those that are between 500 and 1000 warnings will be displayed. Numbers smaller than 500 indicate serious errors.

3.5 Program execution time measurement

The program execution time is measured, displayed on screen and written in the log file ”deeco.log”. ”User time” is the CPU time in seconds spent executing user code; ”Real time” is equivalent to the ”wall-clock-time” difference between program start and stop.

A Libraries

A.1 Standard HP-UX libraries (System services and C functions)

(see (lrom) ©HP-UX Reference)

A.2 HP C++ libraries (Stream Library)

(see (lrom): ©HP C++ Programmer’s Guide)

A.3 Standard Components Library

(see (lrom): ©HP C++ Programmer’s Guide)

The USL C++ Standard Components library contains many general-purpose classes you can use in your applications, including:

- dynamic arrays
- graphs
- lists
- memory allocation

- sets
- bags
- strings

The header files for the Standard Components library are in the directory /usr/include/SC. Use -I/usr/include/SC to direct the compiler to search these header files.

A collection of program development tools for use with the Standard Components library are in /usr/bin. These include hier, incl, publik, dem, and g2++comp.

For more information about the Standard Components, refer to the USL C++ Standard Components Manual. To see the online manual pages, first add the directory /usr/CC/man/SC to your MANPATH environment variable. Then type man name where name is a particular manual page. For an introduction to Standard Components, type man SC_intro and man SC_tools_intro.

Use as compiler option: -lGA -lGraph -l++ (in this sequence).