Chapter

8

8. Further Information

Read this chapter to learn where to find further information about modula-3.

This chapter contains list of pointers to other information available about CM3-IDE and Modula-3. Much of the material cited in this chapter is included in your CM3-IDE distribution, whether in print, or on-line.

For continuously up-to-date information, see:

- Critical Mass Modula-3 Home Page, http://modula3.elegosoft.com/cm3/
- Modula-3 Resource Page, http://modula3.org/
- Modula-3 Internet Newsgroup: **comp.lang.modula3**
- Modula-3 Home Page, http://www.cs.arizona.edu/~collberg/Research/Modula-3/modula-3/html/home.html
- HP Labs (formerly Digital Equipment Corporation Systems Research Center [DEC SRC]), archive of Technical Reports, http://www.hpl.hp.com/techreports/Compaq-DEC/?jumpid=reg_R1002_USEN#src

Chapter Organization

Books on page 156 lists some of the introductory and advanced books on Modula-3.

Technical Documentation on page 157 includes references to a number of technical information sources. Much technical documentation is available as part of your CM3-IDE distribution.

If you are new to Modula-3, you may consider reading articles referenced in **Introductory Programming Articles** on page 158.

Systems Built Using Modula-3 on page 160 includes references to some of the systems written in Modula-3.

Parallel Programming on page 161 cites references to articles written about parallel programming in Modula-3.

Garbage Collection on page 161 describes the local and distributed memory management algorithms used in CM3-IDE.

Comparisons to Other Languages on page 162 may be useful if you would like to find out about similarities and differences between Modula-3 and other languages.

8.1 Books

8.1.1 System Programming with Modula-3

Greg Nelson (editor), Prentice Hall Series in Innovative Technology
ISBN 0-13-590464-1, L.C. QA76.66.S87, 1991.

This book is the definitive language reference. It includes the language reference manual and papers on the I/O library, threads, and the Trestle window system. On the newsgroups and in informal discussion it is often referred to as "SPwM3".

Here is the table of contents:

- Introduction
- Language Definition
- Standard Interfaces
- An Introduction to Programming with Threads
- Thread Synchronization: A Formal Specification
- I/O Streams: Abstract Types, Real Programs
- Trestle Window System Tutorial
- How the Language Got its Spots

8.1.2 Modula-3

Samuel P. Harbison
ISBN 0-13-596396-6, Prentice Hall, 1992.

A complete Modula-3 textbook covering the full language, with examples and exercises. Includes a style manual and a user's guide for SRC Modula-3. The first edition of the book contains many typos. A list of errata is available on-line for anonymous FTP (in TeX, compressed PostScript, or DVI format) from gatekeeper.dec.com in the directory pub/DEC/Modula-3/errata/.

8.1.3 Algorithms in Modula-3

Robert Sedgewick
Addison-Wesley
ISBN 0-201-53351-0, L.C. QA76.73.M63S43, 1993.

Sedgewick's classic text on algorithms, with examples in Modula-3.

8.1.4 Programming with Modula-3: An Introduction to Programming with Style

Laszlo Boeszoermenyi and Carsten Weich
577 pages
Springer-Verlag
ISBN 3-540-57912-5 (English version)
ISBN 3-540-57911-7 (German version), 1995.

This book is an introductory programming text that uses Modula-3 for its examples. To quote the authors, "The main concern of the book is to give a clean and comprehensive introduction to programming for beginners of a computer science study. We start with more traditional programming concepts and move toward advanced topics such as object-oriented programming, parallel & concurrent programming, exception handling, and persistent data techniques. The book also presents a large number of complete examples written in Modula-3."

8.2 Technical Documentation

Several technical reports describe various aspects of CM3-IDE. Most of these reports are available on-line in the Technical Notes section of your CM3-IDE environment.

8.2.1 Reactor White Paper

Critical Mass, Inc. August 15, 1996.
http://www.cmass.com/reactor

Reactor combines an innovative application development system with a rich and robust distributed infrastructure. This report outlines the features and benefits of Reactor's high-productivity, distributed application development system.

8.2.2 Some Useful Modula-3 Interfaces

Jim Horning, Bill Kalsow, Paul McJones, Greg Nelson
Systems Research Center, Digital Equipment Corporation
Report #133, December 1993, 103 pages.

This manual describes a collection of interfaces defining abstractions that Modula-3 programmers have found useful over a number of years of experience with Modula-3 and its precursors. We hope the interfaces will be useful as a "starter kit" of abstractions, and as a model for designing and specifying abstractions in Modula-3.

8.2.3 Network Objects

Andrew Birrell, Greg Nelson, Susan Owicki, Edward Wobber
Systems Research Center, Digital Equipment Corporation
Report #115, February 1994.

This report describes the design and implementation of a Modula-3 network objects system, which allows you to write programs that communicate over a network, while hiding the messy details of network programming. Network objects provide functionality similar to remote procedure call (RPC), but they are more general and easier to use. The system is implemented in Modula-3.

8.2.4 Trestle Reference Manual

Mark S. Manasse and Greg Nelson
Systems Research Center, Digital Equipment Corporation
Report #69, December, 1991.

This report is the working definition of the Trestle toolkit for doing graphics in Modula-3.

8.2.5 VBTkit Reference Manual: A toolkit for Trestle

Marc H. Brown and James R. Meehan (editors)
Systems Research Center, Digital Equipment Corporation

This report is the working definition of the VBTkit toolkit. VBTkit is a collection of widgets for building graphical user interfaces in Modula-3. See the FormsVBT Reference Manual below, which describes a system for easily composing these widgets.

This document is available on-line as part of the CM3-IDE distribution.

8.2.6 Obliq-3D Tutorial and Reference Manual

Marc Najork
Systems Research Center, Digital Equipment Corporation
Report #129, December 1994, 110 pages.

This report describes Obliq-3D, an interpreted language based on the Anim3D library for building 3D animations quickly and easily.

8.3 Introductory Programming Articles

8.3.1 Modula-3 Reference and Tutorial

Stephen Schaub

An on-line reference and tutorial, available in Tutorials section of your CM3-IDE Environment.

This tutorial is available on-line from the Modula-3 Home Page.

8.3.2 Net Balance: A Network Objects Example

Farshad Nayeri, March1996

The sources of a simple client and server that use network objects.

A gzipped tar archive of the sources and web page is also available.

This example program is available on-line from the Modula-3 Home Page.

8.3.3 Building Distributed OO Applications: Modula-3 Objects at Work

Michel R. Dagenais
Draft, March 1995.

A draft of a book describing the latest object-oriented techniques for developing large interactive distributed applications. The focus is on the Modula-3 libraries and Network Objects, but the first two chapters give an introduction to object-oriented programming in general, and the object methodologies in particular.

8.3.4 Partial Revelation and Modula-3

Steve Freeman
Dr. Dobb's Journal, 20(10):36-42, October 1995.

This article describes how Modula-3's partial revelations promote encapsulation and code reuse. The article is one of five on "object-oriented programming" contained in the same issue (the other four languages are C++, Ada 95, S, and Cobol'97).

8.3.5 Initialization of Object Types

Greg Nelson
Threads: A Modula-3 Newsletter, Issue 1, Fall 1995.

This article describes the rationale for the way object types are initialized in Modula-3. Modula-3 doesn't have type constructors, but you can specify default values for object fields in the type definition. The article also describes the init method convention.

8.3.6 Trestle Tutorial

Mark S. Manasse and Greg Nelson
Systems Research Center, Digital Equipment Corporation
Report #69, May 1992, 70 pages.

This report is a tutorial introduction to programming with Trestle, a Modula-3 window system toolkit implemented over the X Window System and Microsoft Windows. It assumes that you have some experience as a user of window systems, but no previous experience programming with X or Win32.

This article is available on-line as part of the CM3-IDE package.

8.3.7 Trestle by Example

Ryan Stansifer, October 1994.

An on-line introduction to the Trestle window system.

This tutorial is available in CM3-IDE environment at /tutorial/ui/tutorial.html.

8.4 Systems Built Using Modula-3

8.4.1 The Juno-2 Constraint-Based Drawing Editor

Allan Heydon and Greg Nelson
Systems Research Center, Digital Equipment Corporation
Report #131a, December 1994.

This report describes Juno-2, a constraint-based drawing editor implemented in Modula-3. For more information, see the Juno-2 Home Page at http://www.research.digital.com/SRC/juno-2/.

8.4.2 Zeus: A System for Algorithm Animation and Multi-View Editing

Marc H. Brown
Systems Research Center, Digital Equipment Corporation
Report #129, February 1992.

8.4.3 Writing an Operating System with Modula-3

Emin Gün Sirer, Stefan Savage, Przemyslaw Pardyak, Greg P. DeFouw, and Brian Bershad
November 1995.
http://www.cs.washington.edu:80/research/projects/spin/www/papers/WCS/m3os.ps

Describes the experiences of the SPIN group at the University of Washington using Modula-3 to build a high-performance extensible operating system. Debunks some of the myths surrounding Modula-3 by arguing that the SRC reference implementation introduces some inefficiencies that are not imposed by the Modula-3 language itself.

8.4.4 The Whole Program Optimizer

Amer Diwan
Threads: A Modula-3 Newsletter, Issue 1, Fall 1995.

This article motivates and describes an optimizer for Modula-3 programs based on whole-program analysis. On benchmark programs, up to 50% of method invocations can be converted to direct calls.

8.5 Parallel Programming

8.5.1 An Introduction to Programming with Threads

Andrew D. Birrell
Systems Research Center, Digital Equipment Corporation
Report #35, January 1989.

This paper provides an introduction to writing concurrent programs with threads. A threads facility allows you to write programs with multiple simultaneous points of execution, synchronizing through shared memory. The paper describes the basic thread and synchronization primitives, then for each primitive provides a tutorial on how to use it. The tutorial sections provide advice on the best ways to use the primitives, give warnings about what can go wrong and offer hints about how to avoid these pitfalls. The paper is aimed at experienced programmers who want to acquire practical expertise in writing concurrent programs.

A must-read for anyone programming a concurrent system, this paper is included in the Technical Notes section of the CM3-IDE distribution.

8.5.2 Synchronization Primitives for a Multiprocessor: A Formal Specification

A. D. Birrell, J. V. Guttag, J. J. Horning, R. Levin
Systems Research Center, Digital Equipment Corporation
Report #20, August, 1987, 21 pages.

Formal specifications of operating system interfaces can be a useful part of their documentation. This document illustrates this by documenting the thread synchronization primitives available in Modula-3.

8.6 Garbage Collection

8.6.1 Compacting Garbage Collection with Ambiguous Roots

Joel F. Bartlett
Western Research Laboratory, Digital Equipment Corporation
Report #88/2, February 1988.

This report describes one of the algorithms used as the basis for CM3-IDE's garbage collector.

8.6.2 Distributed Garbage Collection for Network Objects

Andrew Birrell, David Evers, Greg Nelson, Susan Owicki, Edward Wobber
Systems Research Center, Digital Equipment Corporation
Report #116, December 1993.

This report describes a fault-tolerant and efficient garbage collection algorithm for distributed systems. It is the algorithm used to garbage collect Network Objects.

8.6.3 Portable, Mostly-Concurrent, Mostly-Copying Garbage Collection for Multi-Processors

Antony L Hosking
Department of Computer Science
Purdue University
West Lafayette, IN 47907, USA
hosking@cs.purdue.edu

The paper describes the mostly-concurrent collector implemented in CM3, which permits application threads to run concurrently with garbage collection even on multiprocessors.

8.7 Comparisons to Other Languages

8.7.1 A Comparison of Modula-3 and Oberon-2

Laszlo Boeszoermenyi
Compares Modula-3 and Oberon-2, two successors to Modula-2.
Structured Programming, Springer-Verlag, 1993. Pgs 15-22.

8.7.2 A Comparison of Object-Oriented Programming in Four Modern Languages

Robert Henderson and Benjamin Zorn
Technical Report CU-CS-641-93
Department of Computer Science, University of Colorado, 1993

The paper evaluates Oberon, Modula-3, Sather, and Self in the context of object-oriented programming. While each of these programming languages provide support for classes with inheritance, dynamic dispatch, code reuse, and information hiding, they do so in very different ways and with varying levels of efficiency and simplicity. A single application was coded in each language and the experience gained forms the foundation on which the subjective critique is based. By comparing the actual run-times of the various implementations it is also possible to present an objective analysis of the efficiency of the languages. Furthermore, by coding the application using both explicit dynamic dispatch and static method binding, it is possible to evaluate the cost of dynamic dispatch in each language. The application was also coded in C++, thereby providing a well-known baseline against which the execution times can be compared.

8.8 Summary

Modula-3, the core of CM3-IDE, has been in extensive use for over a decade, and many of these experiences have been recorded in various articles, books, technical documents and network postings.

Many of these documents are distributed in Technical Notes or Tutorial sections of your CM3-IDE distribution.

Systems Programming with Modula-3. The definitive book on Modula-3 is Systems Programming with Modula-3, often abbreviated as SPwM3. The language reference section from this book is included as part of CM3-IDE.

Some Useful Interfaces. Also included in your CM3-IDE distribution, describes many of the standard Modula-3 interfaces.

Network Objects. To find out about the operation of Modula-3 Network Object system, see the Network Objects manual.

Language Reference. Included as part of your CM3-IDE distribution, the *Language Reference* provides precise information about the semantics of Modula-3 programs.

Other Information Sources. There are many other sources of information on the Internet:

- Critical Mass Modula-3 Home Page, http://modula3.elegosoft.com/cm3/
- Modula-3 Resource Page, http://modula3.org/
- Modula-3 page on Wikipedia, http://en.wikipedia.org/wiki/Modula-3
- Modula-3 Internet Newsgroup: post to **comp.lang.modula3**.

If you have found a useful publication or information source which is not listed in this chapter, please inform us at m3-support@elego.de.

This page left blank intentionally.