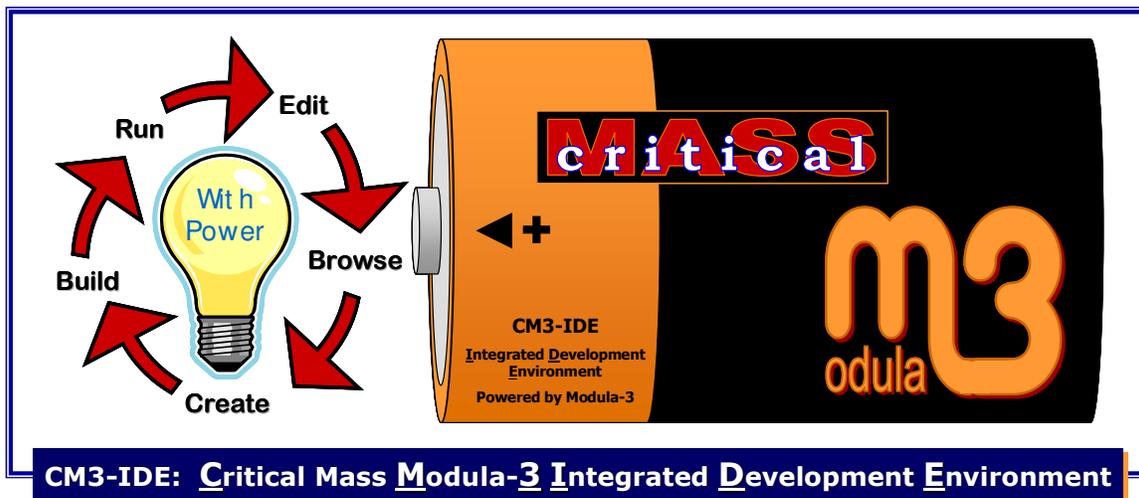




CRITICAL MASS MODULA-3 (CM3)

Integrated Development Environment (IDE)



CM3-IDE User Guide

This page left blank intentionally.

CRITICAL MASS MODULA-3 INTEGRATED DEVELOPMENT ENVIRONMENT

CM3-IDE

User Guide

elego Software Solutions GmbH
Gustav-Meyer-Allee 25 / Building 12
13355 Berlin

June 2008

CM3-IDE was originally developed as Reactor by Bill Kalsow and Farshad Nayeri at Critical Mass, Inc., now named IGEN Corporation. The software was later open-sourced through the tireless efforts of Randy Coleburn and Olaf Wagner.

elego Software Solutions ‘inherited’ the complete sources for the Critical Mass Modula-3 compiler and development system from Critical Mass, Inc., in 2000, and has since made several releases of the system in source and binary form. In March 2002 elego Software Solutions also took over the repository of the other active Modula-3 distribution PM3, till then maintained at the Ecole Polytechnique at Montreal.

So currently elego Software Solutions is hosting the complete CVS source code repositories and providing several possibilities to download Modula-3 sources or installation archives. You must decide if you want sources, CVS repositories (RCS files), or installation archives, and you can get them in several ways: using (anonymous) CVS, CVSup, HTTP, or FTP.

Everybody who wants to work on the CM3 or PM3 sources directly can get write access if he/she provides an ssh key (protocol version 2 (DSA) preferred). Send email to m3-support@elego.de if you are interested, and have a look at the CM3 configuration management rules available on the web site: <http://modula3.elegosoft.com/cm3/>

This manual is derived from the original Reactor User Guide published by Critical Mass, Inc. Copyright © 1996 Critical Mass, Inc. All Rights Reserved.

This manual, as well as the software described in it, is furnished under license and may not be used or copied in accordance with the terms of such license.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Critical Mass, Inc.

The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Critical Mass, Inc., or by elego Software Solutions. Critical Mass, Inc. and/or elego Software Solutions assumes no responsibility or liability for any errors or inaccuracies that may appear in this manual.

Critical Mass, the Critical Mass logo, Reactor, the Reactor logo, CM3, and Critical Mass Modula-3 are trademarks of Critical Mass, Inc.

Alpha AXP and Digital Unix are trademarks of Digital Equipment Corporation. OSF/I is a registered trademark of Open Software Foundation, Inc. Unix is a registered trademark of UNIX System Laboratories, Inc. SPARC, SunOS, and Solaris are trademarks of Sun Microsystems. Microsoft is a registered trademark of Microsoft Corporation. Windows and Win32 are trademarks of Microsoft Corporation. IBM and AIX are registered trademarks of International Business Machines Corporation. HP, HP Precision Architecture, and HP/UX are trademarks of Hewlett Packard. PostScript, and Acrobat are registered trademarks of Adobe Systems Incorporated.

Table of Contents

0.	Introduction	1
0.1	About This Manual	1
0.1.1	Welcome to CM3-IDE!	1
0.1.2	What's Inside	1
0.1.3	Typographic Conventions	2
0.1.4	Before You Begin	2
0.1.5	Keeping in Touch	2
0.2	About CM3-IDE	3
0.2.1	CM3-IDE's Development Environment	3
0.2.2	Systems Development with CM3-IDE	4
0.2.3	Programming in CM3-IDE	4
1.	Learning the Basics	5
1.1	Starting CM3-IDE	6
1.2	A Quick Walkthrough	7
1.3	Creating a Package From Scratch	15
1.3.1	List of All Packages	15
1.3.2	Creating New Packages	17
1.3.3	Procedure Calls in CM3-IDE: What is "IO.Put?"	20
1.3.4	The IO Interface	20
1.3.5	CM3-IDE Makefiles (m3makefile)	24
1.4	Summary	27
2.	The CM3-IDE Environment	29
2.1	Common Tools, Icons, and Visual Elements	29
2.1.1	Quick Access Icons	30
2.1.2	Action Buttons	31
2.1.3	The Find Type-in	32
2.2	CM3-IDE Start Screen	32
2.2.1	Start Screen: System	34

2.2.2	Start Screen: Language	34
2.2.3	Start Screen: Help	34
2.2.4	Customizing the CM3-IDE Start Screen	35
2.3	Summary Screens	36
2.3.1	Package Summary	36
2.3.2	Library Summary	37
2.3.3	Program Summary	39
2.3.4	Interface Summary	40
2.3.5	Module Summary	41
2.4	CM3-IDE's Web Namespace	42
2.4.1	CM3-IDE URLs	43
2.4.2	Example CM3-IDE URLs	43
2.4.3	Regular Expressions in CM3-IDE URLs	44
2.5	Summary	45
3.	<u>Building and Sharing Packages</u>	47
3.1	Building Packages	48
3.2	Directory Structure of a Package	49
3.3	CM3-IDE makefiles	50
3.3.1	Basic Makefile Commands	51
3.3.2	Additional Makefile Commands	53
3.4	Managing Multiple Packages	53
3.5	Shipping Packages	54
3.6	Package Roots	55
3.6.1	Example: Creating a New Package Root	56
3.7	Sharing Packages	57
3.7.1	Example: Adam's and Eve's Joint Project	58
3.8	Builder Options	62
3.9	Summary	64
4.	<u>Customizing CM3-IDE</u>	65
4.1	CM3-IDE Configuration Screen	66
4.1.1	Navigating to the Configuration Page	66
4.1.2	Saving Configuration Changes	66
4.1.3	Display Settings	66
4.1.4	Package Roots Settings	68
4.1.5	Communication Settings	69
4.1.6	Miscellaneous Settings	69

4.1.7	Helper Procedures	70
4.2	Summary	72
5.	<u>Beyond The Basics</u>	73
5.1	Exceptions: Error Handling in CM3-IDE	74
5.1.1	How Exceptions Work	74
5.1.2	Declaring Exceptions	74
5.1.3	Triggering Exceptions: RAISE Statement	75
5.1.4	Handling Exceptions: TRY-EXCEPT Statement	75
5.1.5	Cleaning up: TRY-FINALLY Statement	76
5.1.6	Trapping All Exits from a Block of Code	77
5.1.7	An Example of Exception Handling	77
5.1.8	Programming without Exceptions	77
5.1.9	Making Programs Robust with Exceptions	79
5.2	Object Types: Object-oriented Programming	81
5.2.1	Programming with Objects: A Complete Example	83
5.3	Threads: Managing Concurrent Activities	86
5.4	Opaque Types: Information Hiding And Encapsulation	88
5.4.1	Fully Opaque Types	89
5.4.2	Clients of an Opaque Type	91
5.4.3	Partially Opaque Types: Revealing Types in Moderation	92
5.4.4	Subtyping Partially Opaque Type	94
5.4.5	Clients of a Partially Opaque Type	96
5.5	Generics: Resuable Data Structures and Algorithms	97
5.5.1	Using Generics	98
5.5.2	A Generic Example: List	98
5.5.3	Parameter to a Generic: Atom	99
5.5.4	Instantiating a Generic: AtomList	99
5.5.5	Using Instances of Generics	99
5.5.6	Instantiating Generics for User-Defined Types	101
5.5.7	Instantiating Generics in a Makefile	102
5.6	Unsafe Constructs: System Programming in CM3-IDE	102
5.6.1	Unsafe Coding Example	103
5.7	Summary	105
6.	<u>Development Recipes</u>	107
6.1	Robust Distributed Applications: Network Objects	108
6.1.1	The Common Interface	108

6.1.2	A Network Object Server	109
6.1.3	A Network Object Client	113
6.2	Client/Server Computing: Safe TCP/IP Interfaces	115
6.2.1	A TCP/IP Client: Finger	115
6.2.2	A TCP/IP Server: HTTPD	117
6.3	Taking Persistent Snapshots of Objects: Pickles	120
6.4	Quick Comparison of Large Data: Fingerprints	122
6.5	Portable Operating System Interfaces	124
6.6	Dynamic Web Applications: the Web Server Toolkit	135
6.7	Interacting with C Programs	137
6.7.1	Calling C: A Unix Example	137
6.7.2	Calling C: A Win32 Example	139
6.7.3	Calling Modula-3 from C	140
6.8	Summary	142
7.	<u>CM3-IDE Interface Index</u>	143
7.1	Data Types, Data Structures, and Algorithms	144
7.1.1	Basic Data Types	144
7.1.2	Collections, Lists, Tables, Sets	145
7.1.3	Linked Lists	145
7.1.4	Sorted Linked Lists	145
7.1.5	Property lists	145
Tables		146
7.1.6	Sorted tables	146
7.1.7	Sequences	146
7.1.8	Priority queues	147
7.1.9	Sets	147
7.1.10	Sorting Lists, Tables, and Arrays	147
7.2	Standard Libraries	148
7.2.1	Math, Geometry, Statistics, Random numbers	148
7.2.2	Floating point	148
7.2.3	Environment, Command line parameters	148
7.2.4	I/O streams, Reading and Writing, Files	148
7.2.5	Formatting, I/O Conversion	149
7.2.6	Threads	149
7.3	Systems Development	150
7.3.1	Distributed and Client/Server Development	150
7.3.2	Databases and Persistence	150

7.3.3	Operating System, Files, Processes, Time	151
7.3.4	Interoperability with C	153
7.3.5	Low-level Run-time Interfaces	153
7.4	Miscellaneous	153
8.	Further Information	155
8.1	Books	156
8.1.1	System Programming with Modula-3	156
8.1.2	Modula-3	156
8.1.3	Algorithms in Modula-3	156
8.1.4	Programming with Modula-3: An Introduction to Programming with Style	157
8.2	Technical Documentation	157
8.2.1	Reactor White Paper	157
8.2.2	Some Useful Modula-3 Interfaces	157
8.2.3	Network Objects	158
8.2.4	Trestle Reference Manual	158
8.2.5	VBTKit Reference Manual: A toolkit for Trestle	158
8.2.6	Obliq-3D Tutorial and Reference Manual	158
8.3	Introductory Programming Articles	158
8.3.1	Modula-3 Reference and Tutorial	158
8.3.2	Net Balance: A Network Objects Example	159
8.3.3	Building Distributed OO Applications: Modula-3 Objects at Work	159
8.3.4	Partial Revelation and Modula-3	159
8.3.5	Initialization of Object Types	159
8.3.6	Trestle Tutorial	159
8.3.7	Trestle by Example	160
8.4	Systems Built Using Modula-3	160
8.4.1	The Juno-2 Constraint-Based Drawing Editor	160
8.4.2	Zeus: A System for Algorithm Animation and Multi-View Editing	160
8.4.3	Writing an Operating System with Modula-3	160
8.4.4	The Whole Program Optimizer	160
8.5	Parallel Programming	161
8.5.1	An Introduction to Programming with Threads	161
8.5.2	Synchronization Primitives for a Multiprocessor: A Formal Specification	161
8.6	Garbage Collection	161
8.6.1	Compacting Garbage Collection with Ambiguous Roots	161
8.6.2	Distributed Garbage Collection for Network Objects	162

8.6.3	Portable, Mostly-Concurrent, Mostly-Copying Garbage Collection for Multi-Processors	162
8.7	Comparisons to Other Languages	162
8.7.1	A Comparison of Modula-3 and Oberon-2	162
8.7.2	A Comparison of Object-Oriented Programming in Four Modern Languages	162
8.8	Summary	163

Figures

Figure 1. The Start Screen.....	6
Figure 2. Examples Area.....	7
Figure 3. A Package Summary	8
Figure 4. A Module Summary.....	10
Figure 5. A Schematic of the IO Interface.....	12
Figure 6. Building a package.....	13
Figure 7. A Package Summary containing a Built Program.....	13
Figure 8. A Program Summary with a Run Button	14
Figure 9. Running "Hello World".....	15
Figure 10. Packages Page: Listing of All Packages in CM3-IDE.....	16
Figure 11. Top of the Packages Page.....	17
Figure 12. Package Creation Dialog.....	18
Figure 13. The top of the Module Summary Page.....	19
Figure 14. A Procedure Call Crossing Module Boundaries.....	20
Figure 15. The top of the IO Interface.....	21
Figure 16. Some Procedures Defined in the IO Interface.....	22
Figure 17. The top of a Program Summary page.....	26
Figure 18. Quick Access Icons for <code>/proj/hello/src/hello.m3</code>	30
Figure 19. CM3-IDE's Start Screen	33
Figure 20. A Package Summary	36
Figure 21. A Library Summary.....	38
Figure 22. Top Portion of a Program Summary	39
Figure 23. An Interface Summary	40
Figure 24. Portions of a Module Summary.....	41
Figure 25. CM3-IDE's Build Button	48
Figure 26. CM3-IDE Package Directory Structure	49
Figure 27. Package Roots Section of the Configuration Screen	56
Figure 28. The "Package Roots" section of Eve's Configuration Page.....	59
Figure 29. Save Changes and Display Settings	66
Figure 30. Package Root and Communication Configuration	68
Figure 31. Miscellaneous Settings and Helper Procedures	70

This page left blank
intentionally.

0. Introduction

0.1 About This Manual

0.1.1 Welcome to CM3-IDE!

If you are reading this manual because you would like to learn how to use CM3-IDE, you've come to the right place.

This manual will teach you what you need to know to use CM3-IDE for your development tasks. To make the most of this manual and of CM3-IDE, you should try out the hands-on tutorials using CM3-IDE's development environment.

0.1.2 What's Inside

Chapter 1, **Learning the Basics** on page 5 introduces you to the basic concepts of CM3-IDE: packages, modules, interfaces, importing, and exporting, by walking through two hands-on tutorials.

Chapter 2, **The CM3-IDE Environment** on page 29 tours the commonly used screens in the CM3-IDE development environment.

Chapter 3, **Building and Sharing Packages** on page 47 explains how to build packages, and how to share them with other developers in your team. By the end of Chapter 3, you should have a solid understanding of the CM3-IDE development environment.

Chapter 4, **Customizing CM3-IDE** on page 65 describes CM3-IDE's Configuration page. Use this information to tailor CM3-IDE's behavior to fit your individual or team development needs.

Chapter 5, **Beyond the Basics** on page 73 introduces the more advanced language features, such as object types, threads, exception handling, generics, and provisions for unsafe code.

Chapter 6, **Development Recipes** on page 107 includes recipes for building some simple, but real applications: client/server computing, distributed computing, building



INTRODUCTION

dynamic web applications, integrating legacy code, and using operating system interfaces.

Chapter 7, **CM3-IDE Interface Index** on page 143 outlines the most common of the hundreds of interfaces in CM3-IDE.

Chapter 8, **Further Information** on page **Error! Bookmark not defined.** cites other sources of information about CM3-IDE.

0.1.3 Typographic Conventions

In this manual, the body text is typed in a serif typeface. References to code are typeset a sans serif code typeface, for example:

```
FOR i := 1 TO 10 DO IO.PutInt(i) END;
```

Step

Performing some tasks may involve multiple steps. To help continuity of the text in this manual, each step is marked on the left hand-side with a “STEP” icon.

0.1.4 Before You Begin

To achieve the most from reading this user guide, you should:

- Know the basics of programming. Reading this manual does not require extensive programming experience, however, you are expected to know the basics.
- Learn how to operate your web browser and text editor. CM3-IDE allows you to use a browser and editor of your choice; hence you must know how to use them before you can use CM3-IDE effectively.

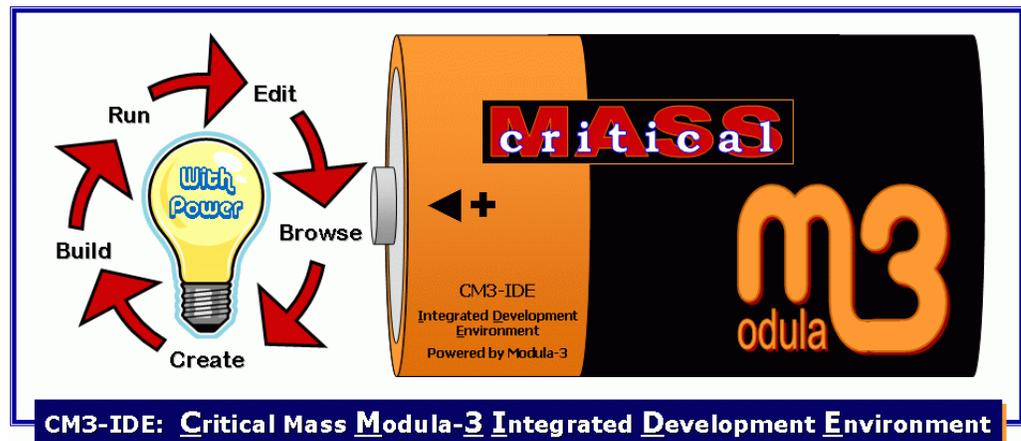
If CM3-IDE’s default browser and editor don’t match your preferences, change the CM3-IDE settings as described in Chapter 4, **Customizing CM3-IDE** on page 65.

- Install CM3-IDE on your system. For instructions, see the *Installation Guide* distributed with your copy of CM3-IDE.

0.1.5 Keeping in Touch

Our web address is <http://modula3.elegosoft.com/cm3/>. Visit us there for updates and information about CM3-IDE. Good luck. We hope you enjoy using CM3-IDE!

0.2 About CM3-IDE



CM3-IDE is a complete environment for the development of robust, multi-platform, client/server and distributed applications.

CM3-IDE provides built-in support for modern systems programming tasks through the use of features such as multi-threading, automatic garbage collection, exceptions, and separation of interfaces from implementations.

CM3-IDE's strength lies in its support for the development of high-performance back-end servers and middle layers in a multi-tier architecture. You can also use CM3-IDE to build user interfaces.

0.2.1 CM3-IDE's Development Environment

CM3-IDE development environment is a dynamic, custom web server. To navigate within the environment, you use a standard web browser such as Netscape Navigator or the Internet Explorer.

Each file in your project and every command in CM3-IDE maps to a location in the CM3-IDE's web namespace. CM3-IDE continually tracks changes in your system and uses the information to build and maintain hypertext links to program components.

This unique design has many advantages, among them:

- The web is an intuitive metaphor to most users, hence it is easy to begin using CM3-IDE.
- CM3-IDE's feel and function remain the same on all platforms.
- Within your projects, you can embed references to external documentation, point out relationships with other projects in your system, or create links to a

INTRODUCTION

README file. You can embed a reference to a CM3-IDE project in an e-mail message.

- CM3-IDE's web-based metaphor allows you to quickly access information about program components, types, and their relationships.

0.2.2 Systems Development with CM3-IDE

CM3-IDE's high-performance compiler generates native code from the same source code whether you run on Windows or Unix. The smart but simple-to-use builder keeps track of dependencies between various program elements automatically, whether or not you use makefiles. Customizing CM3-IDE is straightforward, also.

The CM3-IDE installation includes a collection of portable, well-documented, and thread-friendly libraries, giving you access to thousands of calls in several hundred interfaces.

CM3-IDE's simple repository system allows you and your co-workers to share released code.

0.2.3 Programming in CM3-IDE

CM3-IDE's primary programming language is Modula-3.

Modula-3's Pascal-like syntax and concise definition make it easy to learn and use; yet Modula-3 is an extremely powerful and versatile development tool. Modula-3 has been used extensively for building robust, distributed programs for over a decade.

A language reference, tutorial, and many examples are available online as part of the CM3-IDE environment.