Contents

About This Manual vii
Customer Support viii

Chapter 1 Overview 1

1.1 Why Use the Converter? 1

1.2 A Simple Conversion 3
1.2.1 Invoking the Converter 3
1.2.2 The Converter's Action 6
1.2.3 Is BART Typical? 7

1.3 Writing Scalable Fortran 9

1.4 What Does the Converter Do? 20
1.4.1 Array Operations 20
1.4.2 Array Homes 29
1.4.3 Argument Passing 32

Chapter 2 The Conversion Process 37

2.1 Invoking the Converter 37
2.1.1 Quick Forms 38
2.1.2 Specifying Options 38
2.1.3 Getting Information 38
2.1.4 Creating and Manipulating Packages (Without Translation) . . 40
2.1.5 Translating packages 40
2.1.6 Removing Package Debris 40

2.2 Controlling Conversion Output and Input 42
2.2.1 Output Decisions 42
2.2.2 Getting Conversion Information 43
2.2.3 Input Decisions 44

2.3 Approaches to a Conversion Process 45
2.3.1 Performing Conditional Conversion 45
2.3.2 Using make Files 46
2.3.3 Converting Partial Programs with Firewalls 48
2.3.4 Converting Partial Programs without Firewalls 58

Copyright © 1994 Thinking Machines Corporation

2.4 Controlling Conversion Rules with Options 60
2.4.1 Controlling Vectorization on Small Arrays or Loops 60
2.4.2 Controlling Code Restructuring 60
2.4.3 Controlling Vectorization Analysis 61
2.4.4 Processing Fortran 77 with Array Extensions 61
2.4.5 Accepting CM Fortran Keywords in `.f` Files 62
2.5 Controlling Conversion with Directives 63
2.5.1 `CMAX$` Converter Directives 63
2.5.2 `CMF$` Compiler Directives 66
2.6 The Porting Process 69
2.6.1 Where Does the Converter Fit In? 69
2.6.2 Converting a Program Iteratively 72

Chapter 3 General Porting Issues 73

3.1 Check the Input Program for Correctness 73
3.2 Standardize the Input Program 74
3.2.1 Some Code-Checking Tools 74
3.2.2 Code-Checking on the CM System 76
3.3 Nonstandard Coding Practices 76
3.3.1 Uninitialized Variables 77
3.3.2 Out-of-Bounds Array References 77
3.3.3 Aliasing from Above 79
3.4 Remove Outmoded “Optimizations” 80
3.5 Conditionalize Machine-Specific Code 82
3.6 Miscellaneous Conversion Hints 84
3.6.1 Limitation on Array Promotion 84
3.6.2 Include Files 85
3.6.3 `SAVE` Variables in Procedure Variants 86

Chapter 4 Porting to CM Fortran 89

4.1 CM Array Declarations 90
4.1.1 Array Layouts 90
4.1.2 Arrays in `COMMON` 91
4.2 Customizing I/O Operations 92
4.2.1 I/O into a Single Vector 92
4.2.2 Recoding for Parallel File I/O 92
4.2.3 Block Data Transfers for Serial I/O 93

Version 2.0, May 1994
Copyright © 1994 Thinking Machines Corporation

4.3 Avoiding Memory Model Assumptions 94
4.3.1 Coding around **EQUIVALENCE** 94
4.3.2 Coding for CM Array Argument Passing 97
4.4 Dynamic Array Allocation 99
4.4.1 Simulating Dynamic Allocation in Fortran 77 99
4.4.2 A CM Fortran Solution 102
4.4.3 A Scalable Fortran 77 Solution 104
4.5 Circular Element Shifts 108
4.6 Converting to Nodal CM Fortran with Message Passing 110

Appendixes

Appendix A User Interface Reference 113
A.1 The **cmax** Command 114
A.1.1 Invoking the **cmax** Command 116
A.1.2 **cmax** Translation Options 117
A.2 CMAX Converter Directives 121
A.2.1 Directive Syntax 121
A.2.2 Synopsis of Directives 122
A.3 The CMAX Library 124
A.3.1 Using the CMAX Library 124
A.3.2 The Dynamic Allocation Utility 126
A.3.3 The Circular Shift Utility 128
A.4 Gnu EMACS Utilities for CMAX (Unsupported) 129
A.4.1 The CMAX Viewer 129
A.4.2 Command **cmax-subprogram** 131
A.4.3 Command **cmax-buffer** 131

Appendix B Idioms and Transformations 133
B.1 Fortran 90 and CM Fortran Language Constructs 133
B.1.1 Array Syntax 133
B.1.2 **WHERE** 136
B.1.3 **FORALL** 138
B.2 Fortran 90 and CM Fortran Intrinsic Funtions 142
B.2.1 **ALL** 142
B.2.2 **ANY** 144
B.2.3 **COUNT** 146
B.2.4 **DOTPRODUCT** 148

Copyright © 1994 Thinking Machines Corporation

B.2.5 **MATMUL** 149
B.2.6 **MAXLOC** 151
B.2.7 **MAXVAL** 153
B.2.8 **MINLOC** 155
B.2.9 **MINVAL** 157
B.2.10 **PRODUCT** 159
B.2.11 **SUM** 161
B.3 CM Fortran Utility Library Subroutines 163
B.3.1 **CMF_SCAN_** 163
B.3.2 **CMF_SEND_** 168
B.3.3 **CMF_FE_ARRAY_TO_CM** and **CMF_FE_ARRAY_FROM_CM** . 169

Index

Index **175**

Version 2.0, May 1994
Copyright © 1994 Thinking Machines Corporation