The Message Passing Interface Forum (MPIF), with participation from over 40 organizations, has been meeting since November 1992 to discuss and define a set of library interface standards for message passing. MPIF is not sanctioned or supported by any official standards organization.

The goal of the Message Passing Interface, simply stated, is to develop a widely used standard for writing message-passing programs. As such the interface should establish a practical, portable, efficient, and flexible standard for message passing.

This is the final report, Version 1.0, of the Message Passing Interface Forum. This document contains all the technical features proposed for the interface. This copy of the draft was processed by LaTeX on May 5, 1994.

Please send comments on MPI to `mpi-comments@cs.utk.edu`. Your comment will be forwarded to MPIF committee members who will attempt to respond.

Contents

Acknowledgments **vi**

1 Introduction to MPI **1**

1.1 Overview and Goals . . . 1
1.2 Who Should Use This Standard? . . . 2
1.3 What Platforms Are Targets For Implementation? . . . 3
1.4 What Is Included In The Standard? . . . 3
1.5 What Is Not Included In The Standard? . . . 3
1.6 Organization of this Document . . . 4

2 MPI Terms and Conventions **6**

2.1 Document Notation . . . 6
2.2 Procedure Specification . . . 6
2.3 Semantic Terms . . . 7
2.4 Data Types . . . 8
2.4.1 Opaque objects . . . 8
2.4.2 Array arguments . . . 9
2.4.3 State . . . 9
2.4.4 Named constants . . . 9
2.4.5 Choice . . . 9
2.4.6 Addresses . . . 10
2.5 Language Binding . . . 10
2.5.1 Fortran 77 Binding Issues . . . 10
2.5.2 C Binding Issues . . . 11
2.6 Processes . . . 12
2.7 Error Handling . . . 12
2.8 Implementation issues . . . 13
2.8.1 Independence of Basic Runtime Routines . . . 13
2.8.2 Interaction with signals in POSIX . . . 14

3 Point-to-Point Communication **15**

3.1 Introduction . . . 15
3.2 Blocking Send and Receive Operations . . . 16
3.2.1 Blocking send . . . 16
3.2.2 Message data . . . 16
3.2.3 Message envelope . . . 18
3.2.4 Blocking receive . . . 19
3.2.5 Return status . . . 20

3.3 Data type matching and data conversion 21
3.3.1 Type matching rules 21
3.3.2 Data conversion 24
3.4 Communication Modes 25
3.5 Semantics of point-to-point communication 29
3.6 Buffer allocation and usage 32
3.6.1 Model implementation of buffered mode 34
3.7 Nonblocking communication 34
3.7.1 Communication Objects 36
3.7.2 Communication initiation 36
3.7.3 Communication Completion 39
3.7.4 Semantics of Nonblocking Communications 42
3.7.5 Multiple Completions 43
3.8 Probe and Cancel 48
3.9 Persistent communication requests 52
3.10 Send-receive 56
3.11 Null processes 58
3.12 Derived datatypes 59
3.12.1 Datatype constructors 60
3.12.2 Address and extent functions 67
3.12.3 Lower-bound and upper-bound markers 69
3.12.4 Commit and free 70
3.12.5 Use of general datatypes in communication 72
3.12.6 Correct use of addresses 74
3.12.7 Examples 75
3.13 Pack and unpack 83

4 Collective Communication **90**
4.1 Introduction and Overview 90
4.2 Communicator argument 93
4.3 Barrier synchronization 93
4.4 Broadcast 93
4.4.1 Example using MPI_BCAST 94
4.5 Gather 94
4.5.1 Examples using MPI_GATHER, MPI_GATHERV 96
4.6 Scatter 103
4.6.1 Examples using MPI_SCATTER, MPI_SCATTERV 105
4.7 Gather-to-all 107
4.7.1 Examples using MPI_ALLGATHER, MPI_ALLGATHERV 109
4.8 All-to-All Scatter/Gather 109
4.9 Global Reduction Operations 111
4.9.1 Reduce 111
4.9.2 Predefined reduce operations 112
4.9.3 MINLOC and MAXLOC 114
4.9.4 User-Defined Operations 118
4.9.5 All-Reduce 122
4.10 Reduce-Scatter 123
4.11 Scan 124

4.11.1 Example using MPI_SCAN . . . 124
4.12 Correctness . . . 126

5 Groups, Contexts, and Communicators **130**
5.1 Introduction . . . 130
5.1.1 Features Needed to Support Libraries . . . 130
5.1.2 MPI's Support for Libraries . . . 131
5.2 Basic Concepts . . . 133
5.2.1 Groups . . . 133
5.2.2 Contexts . . . 133
5.2.3 Intra-Communicators . . . 134
5.2.4 Predefined Intra-Communicators . . . 134
5.3 Group Management . . . 135
5.3.1 Group Accessors . . . 135
5.3.2 Group Constructors . . . 136
5.3.3 Group Destructors . . . 140
5.4 Communicator Management . . . 141
5.4.1 Communicator Accessors . . . 141
5.4.2 Communicator Constructors . . . 142
5.4.3 Communicator Destructors . . . 145
5.5 Motivating Examples . . . 146
5.5.1 Current Practice #1 . . . 146
5.5.2 Current Practice #2 . . . 147
5.5.3 (Approximate) Current Practice #3 . . . 147
5.5.4 Example #4 . . . 148
5.5.5 Library Example #1 . . . 149
5.5.6 Library Example #2 . . . 151
5.6 Inter-Communication . . . 153
5.6.1 Inter-communicator Accessors . . . 154
5.6.2 Inter-communicator Operations . . . 156
5.6.3 Inter-Communication Examples . . . 158
5.7 Caching . . . 165
5.7.1 Functionality . . . 165
5.7.2 Attributes Example . . . 169
5.8 Formalizing the Loosely Synchronous Model . . . 171
5.8.1 Basic Statements . . . 171
5.8.2 Models of Execution . . . 172

6 Process Topologies **174**
6.1 Introduction . . . 174
6.2 Virtual Topologies . . . 175
6.3 Embedding in MPI . . . 175
6.4 Overview of the Functions . . . 176
6.5 Topology Constructors . . . 177
6.5.1 Cartesian Constructor . . . 177
6.5.2 Cartesian Convenience Function: MPI_DIMS_CREATE . . . 177
6.5.3 General (Graph) Constructor . . . 178
6.5.4 Topology inquiry functions . . . 180

6.5.5 Cartesian Shift Coordinates . . . 184
6.5.6 Partitioning of Cartesian structures . . . 185
6.5.7 Low-level topology functions . . . 185
6.6 An Application Example . . . 187

7 MPI Environmental Management **189**
7.1 Implementation information . . . 189
7.1.1 Environmental Inquiries . . . 189
7.2 Error handling . . . 191
7.3 Error codes and classes . . . 194
7.4 Timers . . . 195
7.5 Startup . . . 196

8 Profiling Interface **198**
8.1 Requirements . . . 198
8.2 Discussion . . . 198
8.3 Logic of the design . . . 199
8.3.1 Miscellaneous control of profiling . . . 199
8.4 Examples . . . 200
8.4.1 Profiler implementation . . . 200
8.4.2 MPI library implementation . . . 201
8.4.3 Complications . . . 202
8.5 Multiple levels of interception . . . 203

Bibliography **204**

A Language Binding **207**
A.1 Introduction . . . 207
A.2 Defined Constants for C and Fortran . . . 207
A.3 C bindings for Point-to-Point Communication . . . 211
A.4 C Bindings for Collective Communication . . . 214
A.5 C Bindings for Groups, Contexts, and Communicators . . . 215
A.6 C Bindings for Process Topologies . . . 216
A.7 C bindings for Environmental Inquiry . . . 217
A.8 C Bindings for Profiling . . . 217
A.9 Fortran Bindings for Point-to-Point Communication . . . 217
A.10 Fortran Bindings for Collective Communication . . . 221
A.11 Fortran Bindings for Groups, Contexts, etc. . . . 223
A.12 Fortran Bindings for Process Topologies . . . 224
A.13 Fortran Bindings for Environmental Inquiry . . . 225
A.14 Fortran Bindings for Profiling . . . 226

MPI Function Index **227**

Acknowledgments

The technical development was carried out by subgroups, whose work was reviewed by the full committee. During the period of development of the Message Passing Interface (MPI), many people served in positions of responsibility and are listed below.

- Jack Dongarra, David Walker, Conveners and Meeting Chairs
- Ewing Lusk, Bob Knighten, Minutes
- Marc Snir, William Gropp, Ewing Lusk, Point-to-Point Communications
- Al Geist, Marc Snir, Steve Otto, Collective Communications
- Steve Otto, Editor
- Rolf Hempel, Process Topologies
- Ewing Lusk, Language Binding
- William Gropp, Environmental Management
- James Cownie, Profiling
- Tony Skjellum, Lyndon Clarke, Marc Snir, Richard Littlefield, Mark Sears, Groups, Contexts, and Communicators
- Steven Huss-Lederman, Initial Implementation Subset

The following list includes some of the active participants in the MPI process not mentioned above.

Ed Anderson	Robert Babb	Joe Baron	Eric Barszcz
Scott Berryman	Rob Bjornson	Nathan Doss	Anne Elster
Jim Feeney	Vince Fernando	Sam Fineberg	Jon Flower
Daniel Frye	Ian Glendinning	Adam Greenberg	Robert Harrison
Leslie Hart	Tom Haupt	Don Heller	Tom Henderson
Alex Ho	C.T. Howard Ho	Gary Howell	John Kapenga
James Kohl	Susan Krauss	Bob Leary	Arthur Maccabe
Peter Madams	Alan Mainwaring	Oliver McBryan	Phil McKinley
Charles Mosher	Dan Nessett	Peter Pacheco	Howard Palmer
Paul Pierce	Sanjay Ranka	Peter Rigsbee	Arch Robison
Erich Schikuta	Ambuj Singh	Alan Sussman	Robert Tomlinson
Robert G. Voigt	Dennis Weeks	Stephen Wheat	Steven Zenith

The University of Tennessee and Oak Ridge National Laboratory made the draft available by anonymous FTP mail servers and were instrumental in distributing the document.

MPI operated on a very tight budget (in reality, it had no budget when the first meeting was announced). ARPA and NSF have supported research at various institutions that have made a contribution towards travel for the U.S. academics. Support for several European participants was provided by ESPRIT.