1. Extending Ptolemy — Introduction 1-1

1.1 Introduction 1-1

1.2 File Organization 1-1

Ptolemy environment variables 1-2
Directory Structure 1-3

1.3 Creating Custom Versions of pigiRpc 1-6

Creating a pigiRpc that includes your own stars 1-7
Creating a pigiRpc with more extensive customizations 1-8

1.4 Using mkPtolemyTree to create a custom Ptolemy trees. 1-9

mkPtolemyTree example 1-9
How mkPtolemyTree works 1-10
Combining mkPtolemyTree and pigiExample 1-11
Known Bugs in mkPtolemyTree 1-11

1.5 Using csh aliases to create a Parallel Software Development Tree 1-12

Aliases for Managing Symbolic Links 1-12
Creating a Duplicate Hierarchy 1-16
Source Code Control 1-18

1.6 Building standalone programs that use Ptolemy libraries.1-19

Standalone example using StringList 1-19
Standalone example that tests a Scheduler 1-20

1.7 Debugging Ptolemy and Extensions Within Pigi 1-21

A quick scan of the stack 1-22
More extensive debugging 1-23
Debugging hints 1-25

2. Writing Stars for Simulation 2-1

2.1 Introduction 2-1

2.2 Adding stars dynamically to Ptolemy 2-1

2.3 The Ptolemy preprocessor language (ptlang) 2-3

Invoking the preprocessor 2-4
An example 2-4
Items that appear in a defstar 2-5

2.4 Writing C++ code for stars 2-16

The structure of a Ptolemy star 2-17
Reading inputs and writing outputs 2-17
States 2-21
Array States 2-23

2.5 Modifying PortHoles and States in Derived Classes. . . . 2-26

2.6 Programming examples 2-26

2.7 Preventing Memory Leaks in C++ Code 2-28

3. Infrastructure for Star Writers 3-1

3.1 Introduction 3-1
3.2 Handling Errors 3-1
3.3 I/O Classes 3-2
Extended input and output stream classes 3-2
Generating graphs using the XGraph class 3-3
Classes for displaying animated bar graphs 3-4
Collecting statistics using the histogram classes 3-5
3.4 String Functions and Classes. 3-8
3.5 Iterators. 3-10
3.6 List Classes 3-11
3.7 Hash Tables 3-13
3.8 Sharing Data Structures Across Multiple Stars 3-14
3.9 Using Random Numbers 3-17

4. Data Types..... 4-1
4.1 Introduction 4-1
4.2 Scalar Numeric Types 4-1
The Complex data type 4-1
The fixed-point data type 4-3
4.3 Defining New Data Types 4-14
Defining a new Message class 4-15
Use of the Envelope class 4-17
Use of the MessageParticle class 4-18
Use of messages in stars 4-18
4.4 The Matrix Data Types 4-21
Design philosophy 4-21
The PtMatrix class 4-22
Public functions and operators for the PtMatrix class 4-22
Writing stars and programs using the PtMatrix class 4-29
Future extensions 4-33
4.5 The File and String Types 4-34
The File type 4-34
The String type 4-35
4.6 Writing Stars That Manipulate Any Particle Type 4-35
4.7 Unsupported Types 4-37
Sub-matrices 4-37
Image particles 4-40
“First-class” types 4-41

5. Using Tcl/Tk..... 5-1
5.1 Introduction 5-1
5.2 Writing Tcl/Tk scripts for the TclScript star. 5-1
5.3 Tcl utilities that are available to the programmer 5-6

5.4 Creating new stars derived from the TclScript star 5-11
5.5 Selecting colors 5-12
5.6 Writing Tcl stars for the DE domain 5-12

6. Using the Cluster Class for Scheduling 6-1

6.1 Introduction 6-1
6.2 Basic Classes 6-1
6.3 Galaxies and their relationship to Adjacency Lists 6-1
6.4 Clustering 6-2
Initialization — Flattening the User Specified Graph 6-2
Absorb and Merge 6-3
Cluster Iterator Classes 6-5
6.5 Block state and name scoping hierarchy 6-6
6.6 Resetting an InterpUniverse back to actionList 6-6
6.7 References 6-7

7. SDF Domain 7-1

7.1 Introduction 7-1
7.2 Setting SDF porthole parameters 7-1

8. DDF Domain 8-1

8.1 Programming Stars in the DDF Domain 8-1

9. BDF Domain 9-1

9.1 Writing BDF Stars 9-1

10. PN domain 10-1

10.1 Introduction 10-1
10.2 Processes 10-3
The PtThread Class 10-3
The PosixThread Class 10-4
The DataFlowProcess Class 10-6
10.3 Communication Channels 10-7
PtGate 10-8
PosixMonitor 10-8
CriticalSection 10-8
PtCondition 10-9
PosixCondition 10-9
PNGeodesic 10-10
10.4 Scheduling 10-12
ThreadList 10-12
PNScheduler 10-12
10.5 Programming Stars in the PN Domain 10-15

11. SR domain 11-1
11.1 Introduction 11-1
11.2 Communication in SR 11-1
11.3 Strict and non-strict SR stars 11-2
12. DE Domain 12-1
12.1 Introduction 12-1
12.2 Programming Stars in the DE Domain 12-1
Delay stars 12-2
Functional Stars 12-4
Sequencing directives 12-6
Simultaneous events 12-7
Non-deterministic loops 12-8
Source stars 12-8
12.3 Phase-Based Firing Mode 12-11
12.4 Programming Examples 12-13
Identity Matrix Star 12-13
Matrix Transpose 12-14
13. Code Generation 13-1
13.1 Introduction 13-1
13.2 Writing Code Generation Stars 13-2
Codeblocks 13-3
Codeblocks with arguments 13-5
In-line codeblocks 13-7
Macros 13-8
Assembly PortHoles 13-12
Attributes 13-12
Possibilities for effective buffering 13-14
13.3 Targets 13-16
Single-processor target 13-16
Assembly code streams 13-17
Multiprocessor targets 13-18
13.4 Schedulers 13-20
Single-processor schedulers 13-20
Multiprocessor schedulers 13-21
13.5 Interface Issues 13-25
14. CGC Domain 14-1
14.1 Introduction 14-1
14.2 Code Generation Methods. 14-1
14.3 Buffer Embedding 14-2
14.4 Command-line Settable States 14-3

C code generated to support command line arguments 14-3
Changes in pigiRpc to support command line arguments 14-4
Limitations of command line arguments. 14-5
14.5 CGC Compile-time Speed 14-6
14.6 BDF Stars. 14-6
14.7 Tcl/Tk Stars 14-7
14.8 Tycho Target 14-8

15. CG56 Domain 15-1
15.1 Introduction 15-1
15.2 Data Types. 15-1
15.3 Attributes 15-1
15.4 Code Streams 15-2
Sim56Target Code Streams 15-2
S56XTarget/S56XTargetWH Code Streams 15-2

16. C50 Domain 16-1
16.1 Introduction 16-1
16.2 Data Types. 16-1
16.3 Attributes 16-1
16.4 Code Streams 16-1
16.5 Symbols 16-2
16.6 Reserved Memory 16-2

17. Creating New Domains 17-1
17.1 Introduction 17-1
17.2 A closer look at the various classes 17-2
Target 17-3
Domain 17-3
Star 17-3
PortHole 17-3
Geodesic 17-5
Plasma 17-5
Particle 17-5
Scheduler 17-6
17.3 What happens when a Universe is run 17-6
17.4 Recipe for writing your own domain. 17-9
Introduction 17-9
Creating the files 17-9
Required classes and methods for a new domain 17-9
Building an object directory tree 17-10

INDEX I-1