Contents

✥ **Discovering the Possibilities** ✥✥✥

Table of Contents

Table of Contents iii

Preface xvii

- Preface to the First Edition **xvii**
- Preface to the Second Edition **xviii**

Getting Started 1

- Chapter Overview **1**
- Philosophy Behind this Book **1**
- Using this Book **2**
  - Required Version of IDL **2**
  - Working with Colors in the IDL Session **2**
  - Style Conventions Used in the Book **2**
    - Capitalization **3**
    - Comments **3**
    - Line Continuation Characters **4**
  - IDL Programs and Data Files Used in the Book **4**
    - Installing the Program Files **4**
    - Determining Your IDL Home and Current Directory **4**
    - Downloading the Program Files Used with the Book **5**
    - Make Sure Your Coyote Directory is on the IDL Path **5**
    - Copying the Data Files **5**
  - Obtaining Additional Help **6**
- Working with IDL Commands **6**
  - Anatomy of an IDL Command **6**
    - Positional Parameters **6**
    - Keyword Parameters **7**
    - IDL Procedures and Functions **7**

        - Help with IDL Commands **8**
    - Creating Command Journals **8**
- Creating Variables **9**
    - Variable Attributes Change Dynamically **10**
    - Be Careful With Integer Variables **11**
- Working with Vectors and Arrays **13**
    - Creating Vectors **13**
    - Using Array Subscripts **13**
    - Creating Arrays **14**
        - Accessing Elements in Arrays **14**
        - Extracting Vectors and Subarrays **15**
- Working with IDL Graphics Windows **15**
    - Creating Graphics Windows **15**
    - Determining the Current Graphics Window **16**
        - Making a Graphics Window the Current Graphics Window **16**
    - Deleting Graphics Windows **17**
    - Positioning and Sizing Graphics Windows **17**
    - Bringing a Graphics Window Forward on the Display **17**
    - Putting a Title on a Graphics Window **18**
    - Erasing a Graphics Window **18**

Simple Graphical Displays 19

- Chapter Overview **19**
- Simple Graphical Displays in IDL **19**
- Creating Line Plots **20**
- Customizing Graphics Plots **22**
    - Modifying Line Styles and Thicknesses **22**
    - Displaying Data with Symbols Instead of Lines **23**
        - Displaying Data with Lines and Symbols **23**
        - Creating Your Own Plotting Symbols **24**
    - Drawing Line Plots in Color **24**
    - Limiting the Range of Line Plots **25**
    - Changing the Style of Line Plots **26**
- Plotting Multiple Data Sets on Line Plots **28**
    - Plotting Data on Multiple Axes **29**
- Creating Surface Plots **29**
- Customizing Surface Plots **31**
    - Rotating Surface Plots **31**
    - Adding Color to a Surface Plot **32**
    - Modifying the Appearance of a Surface Plot **34**
- Creating Shaded Surface Plots **34**
    - Changing the Shading Parameters **35**
    - Using Another Data Set For the Shading Values **35**

- Creating Contour Plots 36
- Selecting Contour Levels 38
- Modifying a Contour Plot 39
  - Changing the Appearance of a Contour Plot 39
  - Adding Color to a Contour Plot 41
- Creating a Filled Contour Plot 42
- Positioning Graphic Output in the Display Window 44
  - Setting the Graphic Margins 45
  - Setting the Graphic Position 46
  - Setting the Graphic Region 47
  - Creating Multiple Graphics Plots 47
    - Leaving Room for Titles with Multiple Graphics Plots 48
    - Non-Symmetrical Arrangements with !P.Multi 49
- Adding Text to Graphical Displays 51
  - Finding the Names of Available Fonts 52
  - Adding Text with the XYOutS Command 52
  - Using XYOutS with Vector Fonts 53
  - Aligning Text 54
  - Erasing Text 54
  - Orienting Text 54
- Adding Lines and Symbols to Graphical Displays 55
- Adding Color to Your Graphical Displays 56

Working with Image Data 59

- Chapter Overview 59
- Working with Images 59
  - Displaying Images 60
  - An Alternative Image Display Command 62
  - Scaling Image Data 62
    - Scaling Images into Different Portions of the Color Table 63
    - Displaying 8-Bit Images with Different Color Tables on 24-Bit Displays 64
  - Displaying 24-Bit Images 64
    - Displaying 24-Bit Images on 24-Bit Displays 65
    - Displaying 8-Bit Images on 24-Bit Displays 66
    - Automatic Updating of Graphic Displays When Color Tables are Loaded 66
  - Controlling Image Display Order 68
  - Changing Image Size 68
    - Changing Image Size in PostScript 69
  - Positioning an Image in the Display Window 69
    - Positioning Images with Normalized Coordinates 70
  - Reading Images from the Display Device 72
    - Obtaining Screen Dumps on 8-Bit Displays 72
    - Obtaining Screen Dumps on 24-Bit Displays 73
    - Reading a Portion of the Display 73
    - An Alternative to TVRD 73

Basic Image Processing in IDL **74**
Histogram Equalization **74**
Smoothing Images **75**
Removing Noise From Images **77**
Enhancing the Edges of Images **77**
Frequency Domain Filtering of Images **78**
Building Image Filters **78**
Graphical Display Techniques 81
Chapter Overview **81**
Working with Colors in IDL **81**
Using the Indexed versus the RGB Color Model **82**
Static versus Dynamic Color Visuals **83**
Specifying Colors on an 8-Bit Display **84**
Specifying Decomposed Colors on a 24-Bit Display **84**
Determining if Color Decomposition is On or Off **86**
Obtaining Device Independent Colors **87**
Loading Color Tables on a 24-Bit Display **88**
Obtaining a Copy of the Color Table **89**
Modifying and Creating Color Tables **89**
Saving Your Own Color Tables **91**
Creating Your Own Axis Annotations **92**
Adjusting Axis Tick Intervals **92**
Formatting Axis Annotations **92**
Writing a Tick Format Function **94**
Handling Missing Data in IDL **95**
Setting Up a 3D Coordinate System in IDL **97**
Setting Up a 3D Scatter Plot **98**
Positioning the 3D Axes Through the Origin of a Plot **99**
Combining Simple Graphical Displays **100**
Animating Data in IDL **102**
Setting Up the Animation Tool **103**
Loading the Animation Buffer **103**
Running the Animation Tool **103**
Controlling the Animation **103**
Saving Animation Pixmaps **104**
Animating Other Types of Graphic Data **104**
Gridding Data for Graphical Display **106**
Delaunay Triangulation Method of Gridding **106**
Spherical Gridding of Data **108**
Using the Cursor with Graphical Displays **109**
When Is the Cursor Position Returned? **110**
Which Mouse Button Was Used with the Cursor? **111**
Annotating Graphics Output with the Cursor **111**
Drawing a Box **111**

Using the Cursor with Images 112
Using the Cursor in Loops 113

Erasing Annotation From the Display 114
The “Exclusive OR” Method of Erasing Annotation 114
The Device Copy Method of Erasing Annotation 116
Drawing a Rubberband Box 118
Graphics Window Scrolling 119

Graphics Display Tricks in the Z-Graphics Buffer 120
The Z-Graphics Buffer Implementation 121
A Z-Graphics Buffer Example: Two Surfaces 121
Make the Z-Graphics Buffer the Current Device 121
Configure the Z-Graphics Buffer 122
Load the Objects into the Z-Graphics Buffer 122
Take a Picture of the Projection Plane 122
Display the Result on the Display Device 122
Some Z-Graphics Buffer Oddities 123
Warping Images with the Z-Graphics Buffer 123
Transparency Effects in the Z-Graphics Buffer 125
Combining Z-Graphics Buffer Effects with Volume Rendering 126

Reading and Writing Data in IDL 129

Chapter Overview 129

Opening a File for Reading or Writing 129
Locating and Selecting Data Files 130
Selecting File Names 130
Selecting Directory Names 131
Finding Files 131
Constructing File Names 131
Obtaining a Logical Unit Number 131
Using Logical Unit Numbers Directly 132
Allowing IDL to Manage Logical Unit Numbers 132
Determining Which Files are Attached to Which LUNs 133

Reading and Writing Formatted Data 133
Writing a Free Format File 133
Reading a Free Format File 134
Rules For Reading Free Format Data 134
Examples of Reading and Writing Free Format Files 136
Reading a Simple Data File 136
Writing a Column-Format Data File 137
Reading a Column-Format Data File 137
Creating a Template for Reading Column-Format Data 139
Writing with an Explicit File Format 140
A Few Common Format Specifiers 140
Writing a Comma Separated Explicitly Formatted Data File 141
Reading a Comma Separated Explicitly Formatted Data File 141
Reading Formatted Data From a String 141

Reading and Writing Unformatted Data **142**
Reading an Unformatted Image Data File **142**
Writing an Unformatted Image Data File **143**
Reading Unformatted Data Files with Headers **144**
Problems with Unformatted Data Files **145**
Accessing Unformatted Data Files with Associated Variables **145**
Advantages of Associated Variables **146**
Defining Associated Variables **146**

Reading and Writing Files with Popular File Formats **147**
Querying Image Files for Information **147**
Creating a Graphic Display Program **149**
Creating Color GIF Files **150**
If the Display Depth is Eight **151**
If the Display Depth is Greater than Eight **152**
Writing the GIF File **153**
Reading a GIF File **153**
Creating Color JPEG Files **154**
If the Display Depth is Eight **154**
If the Display Depth is Greater than Eight **155**
Writing the JPEG File **155**
Reading a JPEG File **155**
Creating Color TIFF Files **156**
Reading Dicom Image Files **157**
Using the IDLffDicom Object **157**

Reading and Writing HDF Data 161

Chapter Overview **161**

Why Use the HDF Format? **162**

Primary HDF Data Objects **162**

HDF Application Programming Interface **163**

Working with HDF Files **165**
Opening HDF Files **165**
Closing HDF Files **166**
Determining the Number of Tags in an HDF File **166**

Working with Scientific Data Set HDF Files **166**
Optional Dimension Scales **167**
Optional User-Defined Attributes **167**
Optional Predefined Attributes **167**
Opening HDF Files Containing Scientific Data Sets **169**
Closing HDF Files Containing Scientific Data Sets **169**
Creating or Selecting Scientific Data Sets **169**
Creating a New SDS **169**
Selecting an Existing SDS **170**
Adding Attributes to Scientific Data Sets and HDF Files **170**
Gathering Information about Scientific Data Sets **172**
Gathering SDS Attribute Information **173**

Adding Color Palettes to HDF Files **173**

Examples of Reading and Writing HDF Files **174**

Creating Hardcopy Graphics Output 175

Chapter Overview **175**

Selecting the Graphics Hardcopy Output Device **175**

Configuring the Graphics Hardcopy Output Device **176**
- Determining the Current Device Configuration **176**
- Common Device Command Keywords **177**
- Creating the PostScript File **178**

Sending Graphics to the Hardcopy Device **179**

Printing PostScript Files **180**
- Printing PostScript Files on Computers Running MacOS **181**
- Printing PostScript Files on a Windows Computer **181**

Producing Encapsulated PostScript Output **181**
- Encapsulated PostScript Graphic Preview **182**

Producing Color PostScript Output **182**
- Color and Gray Scale Images in PostScript **183**
  - True-Color Images **183**

Creating Quality Output on PostScript Devices **184**
- Similarities Between the Display and PostScript Devices **184**
- Differences Between the Display and PostScript Devices **185**
  - Problem: PostScript Windows May Have a Different Aspect Ratio **185**
  - Solution: Make the Aspect Ratios of Graphics Windows the Same Size **185**
  - Problem: PostScript Devices Have a Higher Display Resolution **186**
  - Solution: Don’t Use Device Coordinates to Position Graphics **187**
  - Problem: PostScript Devices Can Use Different Display Fonts **187**
  - Solution: Take Care in Designing and Positioning Text **187**
  - Problem: PostScript Devices Use Background and Plotting Colors Differently **189**
  - Solution: Understand How PostScript Handles Background and Plotting Colors **190**
  - Problem: PostScript Devices Often Have More Colors Than the Display Device **191**
  - Solution: Be Sure to Scale Your Data Appropriately in PostScript **192**
  - Problem: PostScript Devices Display Images Differently **193**
  - Solution: Size Images with the TV Command **195**

Calculating PostScript Offsets in Landscape Mode **198**

Configuring the PostScript Device with PSConfig **198**

Configuring and Using the Printer Device **200**
- Positioning Graphics with the Printer Device **202**
  - Outputting Images with the Printer Device **203**
- Loading Colors in the Printer Device **204**
  - Color Loading Work-Arounds **204**

IDL Programming Fundamentals 207

Chapter Overview **207**

Writing an IDL Batch File **207**

Writing a Main-Level IDL Program **208**

Writing an IDL Procedure **209**
- Scope of Procedure and Function Variables **210**
- Creating a Positional Parameter **211**
- Defining Optional or Required Positional Parameters **212**
- Defining a Keyword Parameter **213**
  - Using Keyword Abbreviations **213**
  - Defining Optional Keyword Parameters **214**
  - Is the Keyword Defined? **214**
  - Handling Keywords with Binary Properties **215**
  - Passing Undefined Keywords by Keyword Inheritance **216**
- Creating Output Parameters **217**
  - Passing Information by Reference or by Value **217**
  - Using Keyword Inheritance with Output Parameters **219**
  - Is the Parameter Present? **220**
  - Was the Parameter Used? **220**

Writing an IDL Function **221**
- Square Bracket Notation and Function Calls **222**
  - Reserving Function Names with the Forward_Function Command **223**

Using Program Control Statements **223**
- True and False Expressions in IDL **223**
- Making Multiple Statements Appear As Single Statements **224**
- The IF...THEN...ELSE Control Statement **225**
- The Conditional Expression **226**
- The FOR Loop Control Statement **226**
- The WHILE Loop Control Statement **226**
- The REPEAT...UNTIL Loop Control Statement **227**
- The BREAK Control Statement **227**
- The CONTINUE Control Statement **227**
- The CASE Control Statement **227**
- The SWITCH Control Statement **228**
- The GOTO Control Statement **229**
- Error Handling Control Statements **229**
  - The On_IOError Control Statement **229**
  - The On_Error Control Statement **230**
  - The Catch Control Statement **230**
  - Error Handling Hierarchy **231**
  - Reporting Errors **232**
  - Generating Errors **233**
  - Tracing Errors **234**

Compiling and Running IDL Program Modules **235**
- Rules for Compiling IDL Program Modules Automatically **236**
  - Structuring Program Files **236**
- Special Compilation Commands **237**

Writing an IDL Graphics Display Program 239

- Chapter Overview **239**
- The HistoImage Program **240**
  - Writing the Procedure Definition Statement **240**
  - Writing the Error Handling Code **242**
  - Checking for Positional and Keyword Parameters **242**
    - Checking for the Image Positional Parameter **243**
    - Checking for Keyword Parameters **243**
  - Loading the Program Colors **246**
  - Preparing to Draw the Graphics **246**
    - Calculating Graphic Positions in the Window **246**
    - Changing Character Size According To Window Size **247**
    - Calculating the Image Histogram **248**
  - Drawing the Graphics **248**
    - Drawing the Histogram Plot **248**
    - Drawing the Color Bar **251**
    - Drawing the Image Plot **251**
    - Working Around a Printer Device Bug **252**
  - Compiling and Testing the Program **253**
  - Reviewing the HistoImage Program’s Advantages **253**
    - The HistoImage Program is Device Independent **255**
    - Using HistoImage in a “Smart” Resizeable Graphics Window **256**

Writing a Widget Program 259

- Chapter Overview **259**
- The Structure of Widget Programs **259**
- How Do Widget Programs Respond to Events? **261**
- Writing the Widget Definition Module **261**
  - The Advantage of Mistakes **261**
  - Typing the Code **262**
  - Defining and Creating the Program’s Widgets **263**
    - Creating the Top-Level Base Widget **264**
    - Creating Buttons for the Menu Bar **265**
    - Creating the Graphics Window for the Program **265**
  - Realizing the Widgets on the Display **266**
  - Making the Draw Widget the Current Graphics Window **266**
  - Displaying the Graphics in the Draw Widget Window **266**
  - Storing Information Required to Run the Program **267**
    - Using Pointer Variables **268**
    - Using Pointers in the Info Structure **271**
    - Using Widget User Values to Store Program Information **271**
  - Practicing Good Memory Management **272**
  - Creating the Event Loop and Registering the Program **275**
  - Running the Program **276**

Writing the Event Handler Modules 276
Common Fields in Event Structures 277
Event Handler Functions 278
Associating Event Handlers with Widgets 279
Writing the Quit Button Event Handler 279
Writing the Resizeable Graphics Window Event Handler 280
Writing the Cleanup Procedure 282
Running the Program 283
Recovering From Program Errors 283
Adding Color Protection 284
Saving the Color Vectors 285
Setting Up Keyboard Focus Events 285
Modifying the Histo_GUI_TLB_Events Event Handler 286
Buffering the Graphic Display for Smoother Output 287
Widget Programming Techniques 293
Chapter Overview 293
Adding Image Processing Capability 294
Building the Pull-Down Menu Widgets 294
Writing the Event Handler for the Pull-Down Menu 295
Limitations of the Event Handler as Written 296
Implementing an Undo Capability 298
Adding Color Controls to the Program 301
Building the Pull-Down Menu Widgets 301
Writing the Drawing Colors Event Handler 301
Writing the Image Colors Event Handler 305
Communicating in XColors via Widget Events 306
Importance of Group Leaders 310
Adding File Output Functionality 312
Building the Pull-Down Menu Widgets 312
Writing the File Output Event Handler 313
Creating the GIF File 315
Creating the JPEG File 315
Creating the TIFF File 316
An Alternative Way of Creating GIF, JPEG, and TIFF Files 316
Creating the PostScript File 317
Adding Printer Functionality 320
Creating the Print Pull-Down Menu 320
Writing the Print Event Handler 320
Creating Dialog Form Widgets 325
Chapter Overview 325
Creating a Modal Dialog Form Widget 325

- A Blocking Widget Program **326**
- A Modal Widget Program **327**
- Writing a Modal Dialog Form Widget Definition Module **327**
  - Defining a Modal Top-Level Base **328**
  - Defining Other Widgets **329**
  - Storing Collected Information in Modal Dialogs **330**
  - Creating the Info Structure **331**
  - Creating a Blocked Widget **331**
  - Returning from the Block **331**
- Writing the Modal Dialog Event Handler Modules **332**
  - Error Handling **333**
- Testing the Modal Dialog Form Widget Program **335**
  - Using FSC_InputField for Program Input **335**
- Adding an Open Image Capability to the Histo_GUI Program **337**
  - Adding an Open Button **337**
  - Writing the Open Image Event Handler **338**

Creating a Non-Modal Widget Dialog **340**

- Writing a Non-Modal Dialog Widget Definition Module **341**
  - Notifying Widgets of Program Events **342**
- Writing the Non-Modal Dialog Event Handler Modules **343**
  - Sending Events to Other Widgets **344**
- Testing the Readlmage Program **345**
  - Writing the Read Image Event Handler **345**

Creating Graphics Display Objects 349

Chapter Overview **349**

A Quick Object Overview **349**

- The Idea of Data Encapsulation **350**
- Creating Objects **350**
- Invoking Object Methods **351**
- Destroying Objects **353**

Creating a New Object Class **353**

- Defining the Object Class **353**
  - Structure Review **353**
  - Automatic Structure Definition **355**
  - The BoxImage Class Definition **355**
- Creating Object Lifecycle Methods **356**
  - Creating the Init Method **357**
  - Creating the Cleanup Method **360**
- Creating the Specific Instance of the Object **360**
  - Lifecycle Methods Must Be Defined When the Object is Created **361**
  - Common Problems When Creating Objects **361**
  - Initializing the Object Using Parameters **361**
- Creating the Display Method **362**
- Creating Methods to Set and Get Object Properties **364**
  - Set Property Methods **365**
  - Get Property Methods **367**
- Creating Methods to Work with Colors in Objects **370**

Creating Methods to Extend Object Functionality **372**

Object Inheritance **374**
Defining the Subclass Object **374**
Creating Subclass Lifecycle Methods **376**
Attaching Methods to Superclass Objects **378**
Overriding Superclass Methods **378**
Creating New Methods for the SubClass Object **380**
Object Polymorphism **380**
Testing the HistoImage Object **386**

Appendix A: Widget Event Structures 389

Event Structure Definition **389**
Common Field Definitions **389**

Basic Widget Event Structures **389**
Base Widget Event Structure **389**
Button Widget Event Structure **390**
Draw Widget Event Structure **390**
Droplist Widget Event Structure **390**
Label Widget Event Structure **390**
List Widget Event Structure **390**
Slider Widget Event Structure **390**
Table Widget Event Structure **391**
Character Insertion Event **391**
String Insertion Event **391**
Delete String Event **391**
Text Selection Event **391**
Cell Selection Event **391**
Row Height Changed Event **391**
Column Width Changed Event **392**
Invalid Data Event **392**
Text Widget Event Structure **392**
Character Insertion Event **392**
String Insertion Event **392**
Delete String Event **392**
Text Selection Event **392**

Compound Widget Event Structures **393**
CW_Animate Event Structure **393**
CW_Arcbal Event Structure **393**
CW_BGroup Event Structure **393**
CW_Clr_Index Event Structure **393**
CW_Color_Set Event Structure **393**
CW_DefROI Event Structure **393**
CW_Field Event Structure **393**
CW_Form Event Structure **393**
CW_FSlider Event Structure **393**
CW_Orient Event Structure **394**
CW_PDMenu Event Structure **394**
CW_RGBSlider Event Structure **394**

CW_Zoom Event Structure **394**
FSC_InputField Event Structure **394**

Widget Program Event Structures **394**
XColors Event Structure **394**
ReadImage Event Structure **395**

Other Widget Event Structures **395**
Keyboard Focus Events **395**
Kill Widget Request Events **395**
Widget Timer Events **395**
Widget Tracking Events **395**

Appendix B: Data File Descriptions 397

Appendix C: IDL Program Code 399

IDL Example Programs **399**

BoxImage__Define Object Program **399**

HDFRead Program **406**

HDFWrite Program **407**

HistoImage Program **409**

Histo_GUI Program **412**

HistoImage__Define Object Program **423**

OpenImage Program **428**

ReadImage Program **431**

Index 435