Contents

Preface **1**
About this guide 2
Assumptions 2
Conventions 2
New features for developers 3
Online reference material 4
Reference materials on the Visio 2000 product CD 4
Reference materials on the Visio Web site 5

Chapter 1 **Introduction to *Developing Visio Solutions*** **7**
About Visio solutions 8
Modeling with Visio shapes 8
Field sales automation: an example of a Visio solution 12
Using Visio shapes to create solutions 14
Assembling objects into drawings 14
Shapes as components 15
Using SmartShapes technology to develop shapes 17
Using Automation in a Visio solution 19
Automation and Visio objects 19
Monitoring events and totaling values: an example 20
Planning a Visio solution 23
Planning the development process 23
Planning shapes and stencils 25
Planning templates 26
Automating shapes and templates 27
Integrating a Visio solution with a database 27
Choices for implementing Automation 28

Chapter 2 **Creating Visio shapes** **31**
Visio shape anatomy 32
Closed and open shapes 33
1-D and 2-D shapes 34
Shape handles 34
Shapes in groups 36
Drawing new shapes 37
Using the drawing tools to create shapes 37
Drawing closed shapes 39
Drawing shapes with repeated elements 39
Creating groups 39
Merging shapes to create new ones 40

Importing shapes from other programs 42
Importing graphic images 42
Editing imported metafiles and bitmaps 43
Converting imported metafiles to shapes 43
Converting CAD symbol libraries to shapes 44
Adapting existing Visio shapes 45
Revising existing shapes 45
Revising existing groups 46

Chapter 3 Visio masters, stencils, templates, and documents 49
Creating masters and stencils 50
Creating a stencil 51
Creating masters on stencils 52
Editing masters on stencils 53
Creating templates 54
Creating a template 54
About pages, backgrounds, and layers 56
Opening and saving Visio documents 58
Components of a Visio document 58
Opening a Visio file 59
Choosing the right file type for your solution 60

Chapter 4 Visio formulas 61
The ShapeSheet window 62
Displaying a ShapeSheet window 62
Displaying sections in a ShapeSheet window 64
ShapeSheet sections and what they control 65
Examining a shape in a ShapeSheet window 68
Elements of Visio formulas 70
Entering and editing formulas in a ShapeSheet window 70
Functions and operators in Visio formulas 71
ShapeSheet cell references 72
Rules for cell references in formulas 74
Units of measure in Visio formulas 75
Multidimensional units 76
Specifying units of measure 77
Designing Visio formulas 77
How shapes inherit formulas 77
User-defined cells and “scratch” formulas 78
Protecting formulas 80
Controlling recalculation of formulas 81
When to supplement Visio formulas with Automation 82

Chapter 5 Controlling shape geometry with formulas 83
Shape geometry 84
Describing shapes in a coordinate system 85
Representing shape geometry with formulas 86
Representing a shape’s position on a page 88
Preventing users from moving a shape 90

Controlling how shapes stretch and shrink 90
Height-based formulas: an example 91
Optimizing the arrow example 93
Controlling how shapes flip and rotate 94
How flipping affects a shape 94
How rotating affects a shape 95
Designing shapes that flip and rotate 96
Preventing shapes from flipping and rotating 98
Controlling curves in shapes 99
Using rounded corner styles 100
Understanding arcs 100
Converting line and arc segments 104
Useful arc formulas 104
Optimizing shape geometry 106
Using locks to limit shape behavior 107

Chapter 6 Grouping and merging shapes 109

Groups versus merged shapes 110
Creating and controlling groups 111
Grouping and ungrouping shapes 111
Modifying a group 112
How grouping shapes affects their formulas 112
Controlling the behavior of groups 114
Controlling how groups are selected 114
Defining the resizing behavior of grouped shapes 115
Resizing shapes in only one direction 116
Creating a 3-D box: an example 118
Protecting the formatting of shapes in groups 121
Creating and controlling merged shapes 121
Merging shapes 122
Filling merged shapes 123
Hiding shape geometry 124

Chapter 7 Enhancing shape behavior 125

Making shapes flexible with control handles 126
Adding a Controls section to a shape 126
Defining a control handle 127
Setting a control handle's anchor point 129
Setting a control handle's behavior 130
Shortcut menu commands 132
Defining a shortcut menu command 132
Controlling a shortcut command's appearance on the menu 134
Checking commands on the shortcut menu 134
Dimming a shortcut command on the menu 135
Hiding and showing commands on the shortcut menu 136
Using shortcut commands to change shape geometry: an example 137
How the formulas work 138

Custom properties 139
Using custom properties 139
Defining custom properties 141
Adding custom properties to a master 144
Linking custom properties to a database 144
Event formulas 145
Using cells in the Events section 145
Simulating events with the DEPENDSON function 146
Functions that perform actions 147
Performance considerations for event formulas 148

Chapter 8 1-D shapes, connectors, and glue 149

Understanding 1-D and 2-D shapes 150
Converting 1-D and 2-D shapes 151
1-D shape gallery 152
Creating routable and other 1-D connectors 153
Creating routable connectors 153
Creating other 1-D connectors 155
Creating an angled connector 155
Creating a height-based 1-D shape 157
Controlling how shapes connect 160
Defining a connector's glue behavior 161
Specifying what can be glued 162
Understanding connection points 163
Adding connection points to a shape 166
Naming connection points 168
Designing shapes for the dynamic connector 168

Chapter 9 Designing text behavior 171

About text in shapes and masters 172
Viewing text attributes in the ShapeSheet window 173
Controlling the text block's position 174
Adding control handles that control a text block 174
How text control handles appear in the ShapeSheet window 175
Controlling text in a group 176
Resizing shapes with text 176
Controlling text block size 177
Controlling text block width 177
Controlling text block height 178
Basing shape size on the amount of text 178
Basing shape size on text value 179
Changing the font size as a shape is resized 180
Using the SmartShape Wizard to create text resizing formulas 180
Writing custom resizing formulas 181

Controlling text rotation ... 181
Using the SmartShape Wizard to control text rotation ... 182
Gravity formulas ... 183
Counter-rotation formulas for level text ... 183
Constraining text block size: examples ... 184
Constraining the width of a level text block ... 184
Controlling the width of an offset level text block ... 186
Working with text formulas ... 188
Displaying and formatting formula results ... 188
Displaying a shape's width in different units ... 188
Displaying normalized angular values ... 189
Formatting strings and text output ... 190
Using the format function ... 190
Displaying formatted custom properties ... 191
Protecting text formulas ... 192
Testing text block formulas ... 193

Chapter 10 Managing styles, formats, and colors ... 195

Working with styles in the drawing page ... 196
Understanding styles ... 196
Setting default styles for a drawing ... 197
Creating a new style ... 197
Editing a style ... 199
Guidelines for applying styles to shapes ... 199
Reformatting shapes on the drawing page ... 200
Reformatting masters in a stand-alone stencil ... 200
Reformatting all instances of a master ... 202
Using styles in stencils and templates ... 203
Keeping styles consistent across files ... 203
Using naming conventions for styles ... 204
Guidelines for defining styles ... 204
Protecting local shape formats ... 205
Using the Preserve Local Formatting option ... 205
Using the LockFormat cell and the GUARD function ... 205
Managing color in styles, shapes, and files ... 206
Editing the color palette ... 206
Standardizing color palettes across documents ... 206
Using a formula to define a custom color ... 207
Custom patterns ... 208
Creating a custom pattern ... 209
Developing custom fill patterns ... 211
Fill pattern colors ... 211
Designing tiled patterns ... 212
Developing custom line patterns ... 212
Developing custom line ends ... 215

Chapter 11 Arranging shapes in drawings ... 217

Assigning shapes and masters to layers ... 218
Using layers efficiently ... 218
Assigning masters to layers ... 219

Designing a grid 221
Setting the grid for a template's drawing page 221
Creating masters that work with a grid 221
Using formulas to hold grid information 223
Aligning shapes to guides and guide points 224
Guidelines for using guides or grids 225
Manipulating guides and guide points 225
Guides in a rotated page 226
Grouping guides with shapes 227
Using alignment boxes to snap shapes to a grid 227
Adjusting the size of a shape's alignment box 228
Updating an alignment box 230
Changing the alignment box for 1-D shapes 230
Designing shapes for automatic layout 231
Setting layout options for the page 231
Setting shape and connector behavior 232

Chapter 12 Scaled shapes and measured drawings 235

Choosing an appropriate drawing scale 236
Understanding drawing scale and page scale 237
Factors to consider in choosing a drawing scale 237
Choosing a scale for masters 239
Determining an appropriate scale for a master 239
Setting the scale of a master 241
Creating shapes that never scale 242

Chapter 13 Packaging stencils and templates 243

Designing custom shapes for distribution 244
Shape design process guidelines 244
Shape distribution considerations 245
Testing masters 246
Checking the consistency of masters 246
Checking the master in the master drawing window 247
Testing masters with different page scales 247
Adding Help to masters 249
Associating Help with a master 249
Testing shape Help 251
Finishing and testing a stencil 251
Creating master shortcuts 251
Cleaning up masters in a stencil 252
Cleaning up a stencil file 254
Testing stencils 255
Finishing and testing a template 256
Cleaning up a template 256
Testing a template 257
Installing stencils and templates 259
Moving template files 260
Protecting stencils and templates 260

Chapter 14 Automation and the Visio object model 261
An Automation overview 262
The Visio object model 262
Getting and releasing Visio objects 265
Declaring object variables 265
Accessing Visio objects through properties 266
Referring to an object in a collection 266
Iterating through a collection 267
Releasing an object 268
Using compound object references 269
Restricting the scope and lifetime of object variables 270
Using properties and methods 270
Declaring variables for return values and arguments 270
Getting and setting properties 271
Using an object's default property 272
Using methods 272

Chapter 15 Microsoft VBA programming in the Visio application 273
Using the Visual Basic Editor 274
Starting the Visual Basic Editor 275
Navigating among projects 276
Saving a project 276
Creating a VBA project 278
Inserting modules and class modules into your project 279
Inserting user forms into your project 281
Importing files into and exporting files from your project 282
Using the Visio type library 282
Using the Object Browser 283
Setting references to type libraries 284
Using Visio object types 285
Using the global and ThisDocument objects 286
Using the Visio global object 286
Using the ThisDocument object 288
Running VBA code from the Visio application 289
Handling errors 292
Running the program in the right context 292
Verifying that objects and return values exist 293
Checking for error values 293
Managing a VBA project 294
Removing project items 294
Protecting your code 295
Using the Add-in Manager 295

Chapter 16 Working with Visio Document, Page, and Shape objects 297
Working with Document objects 298
Getting a Document object 298
Getting information about documents 299
Working with styles in a document 300
Creating a style for a document 301
Printing and saving documents 301
Working with Page objects 303
Getting a Page object 303
Getting information about pages 304
Adding pages to a drawing 304
Working with Shape objects 305
Getting a shape object 305
Getting information about a shape 307
Creating and changing shapes 309
Adding text to shapes 311
Getting a shape's text 312
Identifying and applying styles to shapes 312
Preserving local formatting 313
Creating groups from a program 314
Creating masters 315
Creating a simple drawing: an example 316

Chapter 17 Automating formulas 319
Working with formulas in cells 320
Getting a Cell object 320
Changing cell formulas using the Formula property 322
Getting the result of a formula 323
Replacing a formula with a result 324
Overriding guarded formulas 324
Using formulas to move shapes: an example 325
Working with sections and rows 326
Adding sections and rows 326
Adding a Geometry section to a shape: an example 328
Deleting sections and rows 330
Changing the type of a segment 331
Iterating through a collection of sections and rows: an example 332
Working with inherited data 333

Chapter 18 Drawing with Automation 335
Automating drawing with masters 336
Getting the stencil 336
Getting the master 337
Dropping the master on the page 337
Placing shapes in a drawing 339

Working with selected shapes 341
Getting shapes that are selected in a window 341
Adding and removing shapes in selections 343
Selecting and deselecting shapes in a window 343
Performing operations on selected shapes 344
Determining a selection's scope 344
Background pages 345
Creating and assigning background pages 345
Iterating through the Pages collection: an example 346
Setting up pages and backgrounds: an example 346
Changing page settings 348
Layers 349
Identifying layers in a page or master 349
Identifying the layers to which a shape is assigned 350
Assigning shapes to and removing shapes from layers 350
Adding layers to and deleting layers from pages and masters 351
Changing layer settings 351

Chapter 19 Automating connections in a Visio solution 353

Working with a Connect object 354
Getting information from a connected drawing 356
Determining which shapes are connected 356
Determining which parts of shapes are connected 357
Getting the cells in a connection 358
Guidelines for analyzing a connected drawing 359
Iterating through the connections on a page: an example 360
Creating a connected drawing from a program 362
What can be glued to what 363
Gluing with Cell objects 365
Gluing a shape to another shape 365
Connecting shapes in a flowchart: an example 367

Chapter 20 Integrating data with a Visio solution 369

Associating data with shapes using Automation 370
Adding custom property and user-defined rows 370
Generating and using unique IDs 371
Visio properties for storing and retrieving data 373
Writing code to extract data from a Visio drawing 374
Extracting data from a drawing: an example 374
Examining the code for extracting data from a drawing 377
Writing code to create a Visio drawing from data 378
Creating a drawing from data: an example 379
Examining the code for creating a drawing from data 382
Integrating a Visio solution with a database 383

Chapter 21 Handling Visio events 385
An event overview 386
Writing code behind events 387
Handling events fired by ThisDocument 388
Declaring an object variable using the WithEvents keyword 390
Defining a class to receive events 391
Class module that responds to events: an example 394
Visio Event objects 395
Defining your Event object 395
Getting information about an Event object 397
Creating an Event object that runs an add-on 397
Persistence of an Event object that runs an add-on 398
Creating an Event object that sends a notification 399
The VisEventProc procedure: an example 403
Event objects that send notifications: an example 404
Lifetime of an Event object that sends a notification 405

Chapter 22 Customizing the Visio user interface 407
What you can customize 408
Getting a UIObject object 409
About menu objects 410
About accelerator objects 412
About toolbar objects 412
About status bar objects 414
Planning user interface changes 415
Customizing a copy of the built-in Visio UI versus an existing custom UI 415
Controlling the scope of your UI 416
Controlling the persistence of your UI 417
Making user interface changes 418
Getting a MenuSet, ToolbarSet, AccelTable, or StatusBar object 418
ID constants for window contexts 419
Adding a menu and a menu item 420
Adding a toolbar and a toolbar button 423
Setting properties of an item 426
Removing items from a user interface 427
Removing a toolbar item 429
Removing an accelerator 430
Putting custom UI changes into effect 431
Using custom user interface files 432
About Custom.vsu 432
Saving a custom user interface file 432
Loading a custom user interface file 433
Restoring the built-in Visio user interface 434

Chapter 23 Using ActiveX controls in a Visio solution 435
Adding ActiveX controls to a Visio solution 436
Working in design mode 436
Inserting a control in a drawing 436
Setting the tabbing order of controls 438
Using the Visio ambient properties in controls 438
Printing a drawing without its controls 439
Protecting controls from changes 439
Handling a control's events 439
Working with controls at run time 440
About control names 440
Getting a control from the OLEObjects collection 441
Distributing ActiveX controls in a Visio solution 442
ActiveX controls that interact with shapes: an example 443

Chapter 24 Using the Visio Undo manager in your program 447
The Visio Undo manager 448
An Undo/Redo overview 448
How the Visio Undo manager works with an add-on 449
Creating undo scopes in your add-on 450
Creating an undo scope 450
Associating events with an undo scope 451
Creating undo units 452
Creating an undo unit 452
Adding an undo unit in the Visio Undo manager 454
Creating an undo unit that maintains non-Visio data: an example 454

Chapter 25 Packaging a Visio Automation solution 459
Installing a Visio solution 460
Specifying Visio file paths and folders 460
How the Visio application searches file paths 461
Controlling when your program runs 462
Distributing your program 465
Distributing Microsoft VBA programs 465
Drawing file size in a Microsoft VBA solution 466
Using universal names in your solution 467
Important licensing information 467

Chapter 26 Programming the Visio application with Microsoft Visual Basic 469
Getting a Visio instance 470
Creating an Application object 470
Getting an Application object 471
Releasing an Application object 471
Using the Application object in a Microsoft Visual Basic program: an example 471
Shortcuts for getting a Visio instance 473
Working with an instance's window handle 474
Interacting with other programs 474
Creating a Visio document 475
Handling errors in Microsoft Visual Basic 476

Interpreting the command string the Visio application sends to your program 477
Running the program from the Macros submenu 477
Running the program when a formula is evaluated 477
Running the program with arguments 479
Running the program from the Startup folder 479
Parsing a command string 479
Using the Visio type library in Microsoft Visual Basic projects 480
Migrating from Microsoft Visual Basic to VBA 481

Chapter 27 Programming the Visio application with C++ 483
How the Visio application exposes objects 484
C++ support in the Visio product 485
Using the wrapper classes 486
The interfaces behind the wrappers 488
Obtaining a Visio Application object 490
Values returned by Visio methods 490
Arguments passed to Visio methods 492
Handling Visio events in C++ programs 494
Implementing a sink object 494
Using CVisioAddonSink 495
Visio libraries 497
Advantages of Visio libraries 497
The architecture of a Visio library 498
Declaring and registering add-ons 499
Running an add-on 501

Appendix A Properties, methods, and events by object 505

Appendix B ShapeSheet section, row, and cell indexes 521
Section, row, and cell indexes for shapes 522
Section, row, and cell indexes for styles 527
Section, row, and cell indexes for pages 527
Section, row, and cell indexes for documents 529
Tab cells and row types 529

Glossary 531

Index 545