Mastering ENVY/Developer
Advances in Object Technology Series

Dr. Richard S. Wiener
Series Editor and Editor

Journal of Object-Oriented Programming
101 Communications, Inc.
New York, New York

and

Department of Computer Science
University of Colorado
Colorado Springs, Colorado

1. Object Lessons: Lessons Learned in Object-Oriented Development Projects • Tom Love
2. Objectifying Real-Time Systems • John R. Ellis
3. Object Development Methods • Edited by Andy Carmichael
4. Inside the Object Model: The Sensible Use of C++ • David M. Papert
5. Using Motif with C++ • Daniel J. Bernstein
6. Using CRC Cards: An Informal Approach to Object-Oriented Development • Nancy M. Wilkinson
7. Rapid Software Development with Smalltalk • Mark Lorenz
8. Applying OMT: A Practical Step-By-Step Guide to Using the Object Modeling Technique • Kurt W. Derr
9. The Smalltalk Developer’s Guide to VisualWorks • Tim Howard
10. Objectifying Motif • Charles F. Bowman
11. Reliable Object-Oriented Software: Applying Analysis and Design • Ed Seidewitz and Mike Stark
12. Developing Visual Programming Applications Using Smalltalk • Michael Linderman
13. Object-Oriented COBOL • Edmund C. Arranga and Frank P. Coyle
15. Object Modeling and Design Strategies: Tips and Techniques • Sanjiv Gossain
16. The VisualAge for Smalltalk Primer • Liwu Li
17. Java Programming By Example • Rajiv Sharma and Vivek Sharma
18. Rethinking Smart Objects: Building Artificial Intelligence with Objects • Daniel W. Rasmus
19. The Distributed Smalltalk Survival Guide • Terry Montlick
20. Java for the COBOL Programmer • E. Reed Doke, Ph.D., and Bill C. Hardgrave, Ph.D.
21. Building Application Servers • Rick Leander
22. Mastering ENVI/Developer • Joseph Pelrine, Alan Knight, and Adrian Cho

Additional Volumes in Preparation
Mastering ENVY/Developer

Joseph Pelrine
Alan Knight
Adrian Cho
Contents

Foreword ................................................................. xv
Acknowledgments .............................................. xix
Introduction ....................................................... xxi

Chapter 1: Getting Started ................................................. 1
Overview ............................................................... 1
  What Is ENVY/Developer? ........................................ 1
  History .............................................................. 2
Architecture ............................................................. 3
  The Repository ...................................................... 3
  Concurrent Development ....................................... 5
  ENVY Concepts .................................................... 5
Installation and Setup ............................................... 6
  The Repository ..................................................... 6
Troubleshooting Installation ...................................... 10
Multiple Images .................................................... 13
Passwords ............................................................. 15
VisualAge Features ............................................... 16
Customizing Setup .................................................. 17
Summary .............................................................. 18

Chapter 2: Basic Concepts ............................................... 21
Questions and Answers ............................................. 21
Software Components .............................................. 22
  Applications ....................................................... 22
  Classes and Extensions ........................................ 23
  Methods ........................................................... 24
  Configuration Maps ............................................. 24
  Prerequisites ..................................................... 25
Prerequisite Enforcement ........................................... 27
Prerequisites and Loading ........................................... 27
Unloading ................................................................. 28
Review and Example .................................................. 29
Version Control .......................................................... 30
Editions ................................................................. 31
Versions ................................................................. 31
Edition Lifecycle ....................................................... 32
Terms .................................................................. 34
Releasing Classes ....................................................... 35
Released versus Loaded ........................................... 36
Developing with Class Editions ................................. 37
Complications ............................................................. 40
Trying It Out ............................................................... 43
Summary ................................................................. 44

Chapter 3: Team Development ........................................ 45
Team Development with Class Editions ......................... 45
Keeping in Sync ......................................................... 46
Team Development Example ...................................... 46
The Process of Syncing Up ......................................... 48
Queries ................................................................. 49
Integration ............................................................... 50
Summary ................................................................. 51
One-Click Loading and Application Editions ..................... 52
Application Editions .................................................. 52
Baselining ............................................................... 57
The Baseline Process ................................................. 57
Development During a Baseline ................................... 58
Post-Baseline Steps ..................................................... 58
When to Baseline ....................................................... 59
Component Ownership and Conflict Resolution ............... 60
User Roles ............................................................... 60
Class Owner ............................................................. 61
Resolving Conflicts ................................................... 62
The ENVY Strategy ................................................... 63
Class Ownership Details ............................................. 64
Contents vii

Application Manager .............................................. 65
Configuration Map Manager ........................................ 66
Library Supervisor .................................................. 67
Scratch Editions .................................................... 68
Version Naming Conventions ...................................... 70
Default Version Names ............................................. 70
Extended Version Names .......................................... 70
Version Names for Developer Tools ............................. 71
Summary ............................................................. 71

Chapter 4: Advanced Development ................................. 73
Configuration Management ......................................... 73
Subapplications ..................................................... 74
Conditional Loading of Subapplications ....................... 77
Conditional Loading of Configuration Maps .................... 81
Large Projects ..................................................... 82
Communication ..................................................... 83
Manageability ....................................................... 87
Multiple Projects and Re-Use ..................................... 87
Multiple Streams ................................................... 88
Multi-Site Development .......................................... 89
That’s Nice, But… .................................................... 90
Streams ............................................................. 90
Representing Streams .............................................. 91
Conventions ......................................................... 92
Issues ................................................................. 92
Extreme Programming and ENVY ............................... 93
ENVY Issues ....................................................... 93
Extreme Guidelines ............................................... 94
Odds and Ends ..................................................... 95
Summary ............................................................. 97

Chapter 5: Formal Concepts ....................................... 99
ENVY Products ..................................................... 99
Components ........................................................ 99
Applications ........................................................ 100
Subapplications .................................................... 100
# Classes and Class Extensions
- 100

# Methods
- 101

# Configuration Maps
- 101

# Managing Components
- 102
  - Configuration Management
  - 102
  - Version Control
  - 103

# The Image and the Library
- 105
  - Loading Components
  - 106
  - Scratch Editions
  - 106

# Cooperative Development
- 107
  - Releasing
  - 107
  - Class and Method Editions
  - 107
  - Reference Models
  - 108
  - Continuous Integration
  - 109

# User Roles
- 110
  - Ownership
  - 110
  - Roles in Detail
  - 111

# Development Process
- 114
  - Structuring Components
  - 114
  - Assigning Roles
  - 115
  - Beginning a Development Cycle
  - 115
  - Development
  - 116
  - Baselining
  - 118
  - Multiple Streams
  - 120
  - Code Reviews
  - 120

# Libraries
- 121
  - Library Growth
  - 121

# Summary
- 123

## Chapter 6: Packaging and Delivery
- 125

### Prerequisites Revisited
- 125

### Prerequisites
- 126

### Class Visibility
- 126

### Setting Up Prerequisites
- 127

### Prerequisite Problems
- 129

### Delivering
- 139
## Contents

```
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delivering in Smalltalk</td>
<td>140</td>
</tr>
<tr>
<td>VisualAge Packaging</td>
<td>148</td>
</tr>
<tr>
<td>Packager-Friendly Applications</td>
<td>148</td>
</tr>
<tr>
<td>VisualWorks Packaging</td>
<td>152</td>
</tr>
<tr>
<td>Application Attachments</td>
<td>157</td>
</tr>
<tr>
<td>Version Naming for Developer Tools</td>
<td>159</td>
</tr>
<tr>
<td>Maintaining Consistent Version Numbers</td>
<td>159</td>
</tr>
<tr>
<td>Naming Convention</td>
<td>160</td>
</tr>
<tr>
<td>Post-Release</td>
<td>162</td>
</tr>
<tr>
<td>Incremental Updates and Patches</td>
<td>163</td>
</tr>
<tr>
<td>Summary</td>
<td>163</td>
</tr>
</tbody>
</table>

### Chapter 7: Extending the System

```
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modify State</td>
<td>166</td>
</tr>
<tr>
<td>Loading and Unloading</td>
<td>167</td>
</tr>
<tr>
<td>Initializing Class Variables</td>
<td>167</td>
</tr>
<tr>
<td>Application-Level Hooks</td>
<td>168</td>
</tr>
<tr>
<td>Before and After</td>
<td>169</td>
</tr>
<tr>
<td>Example Utilities</td>
<td>173</td>
</tr>
<tr>
<td>Policy Classes</td>
<td>175</td>
</tr>
<tr>
<td>Changing the Error Reporter</td>
<td>175</td>
</tr>
<tr>
<td>Changing the Default Library Interface</td>
<td>180</td>
</tr>
<tr>
<td>The Art of the Class Extension</td>
<td>182</td>
</tr>
<tr>
<td>Class Extensions</td>
<td>182</td>
</tr>
<tr>
<td>Advanced Class Extensions</td>
<td>183</td>
</tr>
<tr>
<td>Other Techniques</td>
<td>184</td>
</tr>
<tr>
<td>Automating Subapplication Release</td>
<td>186</td>
</tr>
<tr>
<td>An Example — The Menu Extension API</td>
<td>191</td>
</tr>
<tr>
<td>Summary</td>
<td>196</td>
</tr>
</tbody>
</table>

### Chapter 8: Administration

```
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance Operations</td>
<td>197</td>
</tr>
<tr>
<td>Managing Users</td>
<td>198</td>
</tr>
<tr>
<td>ENVY Users</td>
<td>198</td>
</tr>
<tr>
<td>Administering Users</td>
<td>199</td>
</tr>
<tr>
<td>Updating Users</td>
<td>200</td>
</tr>
</tbody>
</table>
```
Transferring Code .................................................. 200
File-In/File-Out .................................................... 201
Managing Libraries ............................................... 206
  Purge ................................................................. 207
  System Upgrades ................................................. 211
  Remote Work ...................................................... 212
Writing Scripts ...................................................... 213
  Prompter Utilities ................................................. 214
  ENVY Protocol .................................................... 215
  Adding a User ..................................................... 215
Organizing for Re-Use ............................................. 217
  The Ideal of Re-Use ............................................. 217
  Economics and Culture ......................................... 219
  Re-Use Mechanisms ............................................. 220
  Achieving Re-Use ............................................... 221
Tips and Tricks ..................................................... 225
  Fragmentation on NTFS ......................................... 225
  Managing Import/Export Directories ....................... 225
Summary ............................................................. 226

Chapter 9: Goodies .................................................. 227
User Fields .......................................................... 228
  Types of User Fields ........................................... 228
  User Fields Protocol ........................................... 228
  User Field Usage ............................................... 230
A Script Manager .................................................... 230
  Basic Operations ................................................ 232
Summary ............................................................. 234
ENVY Programming Concepts ................................... 235
The Browsers ........................................................ 235
  Timestamps, Versions, and Editions ....................... 236
  Application Classes ............................................ 236
  Shadows ............................................................. 237
Three-Way Differences ............................................ 246
  What Do You Mean, Three-Way Differences? ............. 247
  Representing Differences ..................................... 248
Contents

Library Specs (Aside) .................................................. 250
Difference Items ....................................................... 254
Computing Differences ............................................... 255
The Browser ............................................................ 260
Summary .................................................................. 261
A Simple Project Management Tool .............................. 262
MedProject ............................................................... 262
Generating the Methods ............................................. 264
Loading the Projects .................................................. 265
Extracting Information from Projects ......................... 266
Summary .................................................................. 267
Checkpoints ................................................................ 267
Rationale ................................................................ 267
Usage .................................................................... 269
Checkpointing Issues ................................................ 269
Implementation ........................................................ 272
Checkpoint Summary .................................................. 279
ENVY-izing the Refactoring Browser ......................... 279
Providing Functionality in the ENVY Browsers .......... 280
Correcting Functionality .......................................... 280
Renaming Versions .................................................... 281
Enhancing the Menus ................................................. 282
Renaming Application Versions ................................ 283
Renaming Class Versions ......................................... 284
Renaming Configuration Map Versions ..................... 285
Removing Source Code ............................................ 286
Mechanics ............................................................... 286
Summary .................................................................. 289

Chapter 10: Troubleshooting ........................................ 291
Component Loading ................................................... 291
Image Recovery ......................................................... 292
Image Crash Recovery .............................................. 292
Image Crash Recovery (Part II) ................................. 295
ENVY Crash Recovery .............................................. 296
Making Recovery Easier ............................................. 298
Repository Recovery .................................................. 299
Basic Crash Recovery ............................................ 299
Recovering Code ..................................................... 300
Project Recovery ....................................................... 301
Coding Problems .................................................... 301
Integration Problems ............................................. 303
Packaging Problems .................................................. 307
Quality Problems .................................................... 310
Summary .............................................................. 311

Appendix: A Selected Annotated API
of ENVY System Classes ........................................ 313
Application and SubApplication ................................ 314
Protocols ............................................................... 315
Classes ................................................................. 315
Class Visibility ......................................................... 316
Applications/SubApplications .................................. 317
Configuration Expressions ........................................ 318
Editions/TimeStamps ............................................... 318
Menus ................................................................. 319
Pre/Post Loading/Removing ...................................... 319
Shadows ............................................................... 320
Startup/Shutdown ..................................................... 320
User Fields/Library Objects ...................................... 321
Users ................................................................. 322
Version/Edition Status ............................................. 323
EmConfigurationMap .............................................. 323
Protocols ............................................................... 323
EmConfigurationMap Class ....................................... 324
Accessing ............................................................. 324
Applications .......................................................... 325
Configuration Expressions ....................................... 325
Differencing .......................................................... 326
Loading .............................................................. 326
Required Maps ....................................................... 326
User Fields ............................................................ 327
Contents xiii

Users ................................................................. 328
Version/Edition Status ............................................ 328
EmUser Class ......................................................... 328
Class and CompiledMethod ....................................... 329
  Class .................................................................... 329
  CompiledMethod .................................................... 330
Helper Classes ......................................................... 331
  ErTools ............................................................ 331
  EmInterface ....................................................... 332
  System ............................................................ 332

Glossary ............................................................... 333

References ............................................................ 339

Index ................................................................. 341
Foreword

This book describes ENVY/Developer and its use by three very experienced professional software developers who have used it to build both successful products, including Top Link and VA/Java, and applications for numerous customers. Their unique user-oriented perspectives give developers and managers insight into how to use the product successfully in team development projects.

ENVY/Developer began as Orwell (OOPSLA reference), a research prototype developed in the Object Oriented Research Group, at Carleton University. It was built as part of the Actra project; a multiprocessor embedded Smalltalk, funded by the Defense Research Establishment Ottawa. Orwell addressed the problem of coordinating multiple developers working in the OO RAD environment of Smalltalk.

In designing Orwell, we sought to address the needs of our funding agency DREQ, in particular Dr. Brian Barry who would later become CTO, then CEO of Object Technology International (OTI) and was interested in versioning configuration management for Smalltalk. Our goal was to meet the needs of our development team, which meant preserving the incremental nature of the Smalltalk programming environment.

What we developed at Carleton, then quickly re-implemented at OTI, was a strong specific solution for Smalltalk. Little did we know at the time how strong and specific it was to be! It was developed as an internal tool to support building embedded systems, hence the strict rules for compatibility, which were designed to reduce the downstream problems of the packaging or roaming engineer.

ENVY/Developer evolved to meet the needs of software engineers both inside and outside OTI. Although designed for internal use, Orwell escaped under customer pull to become the widely used ENVY/Developer product first for OTI's embedded ENVY/Smalltalk, then Digitalk's Smalltalk/V family, then PPS VisualWorks, followed by IBM's VisualAge for Smalltalk Team, and finally VisualAge for Java Enterprise Edition. It was also customized and built into numerous CASE and test environments. It provides a unique, although sometimes controversial approach to addressing the problem of team development and versioning an incremental object-oriented environment.

Kent Johnson implemented the initial design, and became the lead developer for the product. Kent translated my white board dialogs into working code initially in Smalltalk/V and then V286. During the early years he not only developed ENVY/Manager he managed ENVY/Smalltalk image. We introduced what at that
time were some simple packaging notions called applications, and prerequisites, which are now, of course, as commonplace as packages in Java or assemblies in .Net. Soon Kent moved to OTI, Mike Milinkovich joined him, they introduced the concepts of configuration maps (known more commonly today as manifests) and sub-applications, which allowed the isolation of platform specific code. In later years, Adrian Cho, one of the co-authors of this book, made substantial contributions in numerous areas such as required maps, an ENVY API, and almost single-handedly supporting the VisualWorks platforms, as well as far too many EMR platform and VA/Java.

Pete Lord of NCR, was one of the first to appreciate that ENVY could be extended to support non-Smalltalk artifacts such as test scripts and was in a large part responsible for pulling ENVY/Developer out of OTI engineering. The initial implementation used a C Btree library; the library was as large as the Smalltalk image! In order to satisfy our commitments to the Tektronix oscilloscope team (our first embedded Smalltalk customer) and NCR, Dave Thomson and I quickly tossed the Btree and designed a simple method-specific, lock-free atomic update protocol, while Kent rewrote the persistent dictionary above Dave. The solution was originally designed to use client side locking because we used NetWare file servers; however, Tektronix was developing using Smalltalk/V on Sun workstations with NFS (stateless is not a feature!). John Duimovich and Dave Thomson hacked together a Unix server for ENVY/Manager called EMR, which supported the same protocols as the client side locks.

ENVY/Developer has an attitude which some may say reflected it's inventor. While most version management systems use a sign in/sign out approach and user conflict/merge tools, ENVY opted instead for a human-centered model of component ownership. This was intended to eliminate the low personal commitment we found when programmers checked out modules with no owner, inserted their fixes or updates, and then released them back into the library with little concern for who these changes impacted. Component ownership, in particular class ownership was advocated in the Orwell paper, and became a controversial matter.

In 1987 when OTI started working with Digitalk, "NV" was the code name for New V, the portable 32 bit, generation scavenging Smalltalk implemented by Mike Wilson and John Duimovich, with assistance from Peter Shipton and Dave Thomson. NV was released by Digitalk as Smalltalk/V Mac and sold by OTI as ENVY/Smalltalk to embedded customers who developed using Sun and Apollo workstations as well as PCs. The words NV made it into the Tektronix contract, and not liking acronyms, I decided in a moment of haste that we were building environments and so the English word “envy” was chosen. Being a small company, we trademarked only ENVY; hence all of the products became ENVY/something... The name ENVY/Developer was cleverly chosen by Mike Milinkovich, who was then product manager. He needed a way to package ENVY/Manager, ENVY/Packager, and ENVY/Stats into a single product that would be easy to market.
In practice ENVY allows many styles. Component ownership is often collective during the evolution of a new component and then becomes more static through the life of a component. We are also not religious about the use of merge tools, which are highly appropriate for distributed development of more loosely connected teams. ENVY/Developer's evolution suffered largely due to the fact that it was the stable foundation for the development of Babel, OTI's multi-platform standards-based library, which allows common native widget GUI code on ParcPlace, Digital, and ENVY/Smalltalk embedded platforms. It also supported the development of IBM Smalltalk, VisualAge, and VA/Java.

Concurrently OTI maintains versions of ENVY/Developer for ParcPlace Smalltalk. Sadly, much of the best developer time went into debugging socket libraries, LAN locking problems and Smalltalk vendor platform portability issues. Likely the best kept secret is that OTI has more criticisms, feature requests, and wishes for ENVY/Developer than any of our customers and users.

ENVY/Developer's stability is a testimony to the dedication of the engineering team; its lack of evolution is a testimony to the futility of our industry where vendors take smart talented people and condemn them to platform hell.

The authors have in-depth experience using ENVY/Developer. They have also developed extension tools and best practices to allow it to be used in different settings and tailored for specific needs. The authors’ experiences span three continents. Alan has worked both as a consultant and product developer with Object People and now Cincom in Canada. Joseph has been involved as a consultant on numerous Smalltalk and Gemstone projects in Europe. Adrian had his own Smalltalk business in Sydney, Australia and later joined OTI as the technical lead for ENVY/Developer. We at OTI appreciate their efforts in describing ENVY/Developer from their perspective. Both novice and expert ENVY/Developer users will benefit from their expert user views.
Acknowledgments

Like any book, this one is not an individual achievement and could not have happened without the help and influence of many people.

We would like to give our most sincere thanks to all of those who helped us. First of all, we thank everyone at Cambridge University Press and SIGS Books, particularly our editor Lothlórien Homet for her tremendous patience and commitment to this project.

Dave Thomas provided the original inspiration for ENVY, founded and led OTI, and of course wrote the Foreword. Without him, none of this could have happened. We'd also like to thank everyone at OTI for producing ENVY, and for their invaluable help and cooperation in getting this book written. The VisualAge team at IBM, the VisualWorks team at Cincom, and all the other Smalltalk vendors helped make our programming lives a little bit brighter.

Eric Clayberg, at Instantiations/Smalltalk Systems, contributed the piece of Chapter 9 on renaming versions and hiding source code, and has generally been a cool guy to exchange ideas with as well as an outstanding contributor to the Smalltalk community. Paul Baumann, at Gemstone, provided some very useful input based on his real-life job using ENVY to delivery high quality, multi-dialect, and -platform software product. Jan Steinman, cofounder of Bytesmiths, was a source of inspiration and helped with the outline. Don Roberts and John Brant created the Refactoring Browser and were very helpful in the port to the ENVY browsers.

We have benefited enormously from the feedback we've received on the book from many sources, both from those who reviewed the text and those we've discussed the issues with. We'd particularly like to single out Stephane Ducasse for his tireless review of the text, and to specifically acknowledge Anthony Lander, Donald Smith, and Didier Besset.

We'd like to thank the Smalltalk user community as a whole for its endurance and support, and we'd particularly like to thank those who ordered the book on faith long before it was finished. There are too many people to mention who pestered us to write and finish this book, and of course we've worked with countless people over the years who have asked for and helped shape the things we've implemented here.

In addition to these people, Joseph would like to specifically thank Alan and Adrian for agreeing to work with him on this project, and also (in alphabetical order) Kent Beck, Stefan Bosshard, Kyle Brown, Thorsten Dittmar, Joachim Geidel, Justin
Hill, Ralph Johnson, Daniel Jutzi, Eliot Miranda, Jeff Odell, John Sarkela, Renate Schirmer, Bobby Woolf, all the guys at Daedalos, Camp Smalltalk, and the cast of thousands who kept screaming "when's it gonna be done?"

Alan would like to thank Kirsten Carlson for her near-infinite patience and love, for putting up with all those times he disappeared to work on the book, and for her help and advice on writing. He would, of course, also like to thank Joseph and Adrian for their work on the book. He would also like to thank everyone at The Object People for the wonderful and supportive environment there, the opportunity to work on ENVY with a wide variety of customers, and for many stimulating discussions on ENVY features and usage. In particular, he'd like to thank Wayne Beaton, Dave Buck, Ron Charron, Martin Kobetic, and Anthony Lander. Finally, he'd like to thank Annick Fron and the European Smalltalk User Group for inviting him to speak and making him come up with a topic where he thought he could contribute something.

Adrian would like to thank Joseph and Alan for inviting him to participate in this project. He would also like to thank Dave Thomas and everyone at OTI for their work on ENVY and for their support and encouragement in writing this book.
Introduction

Smalltalk was first built as a personal computing environment. Most systems at that time were time-sharing, with character-based terminals. In contrast, Smalltalk provided a high-resolution graphical screen and a mouse to maximize the power available to the one user actually sitting at the machine. Smalltalk was designed so that every component in the system was accessible to the user and could be understood by one person.

This was an outstanding vision, and one that remains both important and ahead of the mainstream even today. However, this emphasis on the personal meant that cooperative development issues were less well represented.

Individuals could be incredibly productive, but the code resided in each developer’s image and was not directly visible to other team members. The code could be filed out and exchanged between developers but on an ad hoc basis. The way Smalltalk represented code, and the development practices that Smalltalk encouraged were not a good match for the file-based version control systems of the day. As Smalltalk became more widely used in industry, the need for support of large teams became more critical.

Different schemes were attempted that built relatively lightweight Smalltalk layers on top of existing team programming systems. Finally, in the mid-1980s, Carleton University Professor Dave Thomas and some of his students broke the mold and came up with Orwell, a repository-based system that was tightly integrated with the Smalltalk environment. This was later commercialized as ENVY/Developer, and it rapidly became the dominant team development system for commercial Smalltalk work. In recent years, the same ideas have been the basis for the team environment in IBM’s VisualAge for Java and VisualAge Generator products.

Although ENVY has long been widely used, no books have been available on it. Developers using ENVY were forced to rely on courses, magazine articles, and folklore to understand how to use and extend the system. This book was written to help fill that void. We hope to provide an introduction that lets new users quickly come up to speed and be productive, a reference that will let experienced users and administrators find the information they need, and information to help toolsmiths and power users extend the system in the ways they need. We also hope to provide the community with some useful tools we have built and described in this book.
Structure of This Book

This book is divided into two major sections. The first section (Chapters 1 through 6) is about standard ENVY usage. It introduces ENVY concepts and practices in both a tutorial and more theoretical format. The second section (Chapters 7 through 10) is devoted to particular issues and to extending and enhancing the ENVY system. This section includes many different topics, appropriate for system administrators, advanced developers, and toolsmiths.

Who Should Read This Book

This book will be useful to anyone who works with ENVY, or who is interested in team programming mechanisms, particularly as they are applied to object-oriented languages and interactive development environments. The book is primarily aimed at three audiences.

The first, and perhaps the most important, are those learning ENVY or developing with it on a day-to-day basis. For these readers the tutorial chapters and explanation of ENVY concepts will be most useful. Basic knowledge of Smalltalk is assumed, but no prior knowledge of ENVY is required. For these readers, Chapters 1 through 4 provide an ENVY tutorial with examples and advice. A quicker and more formal summary of ENVY concepts is provided in Chapter 5: “Formal Concepts.” It may also be useful to read Chapter 6: “Packaging and Delivery,” Chapter 8: “Administration,” and Chapter 10: “Troubleshooting,” but these are not necessary for basic ENVY usage. A subset of this audience consists of project managers using ENVY who want to understand the concepts and how they affect work processes but who don't need to use the tool. For these readers the most useful resources will be Chapter 5: “Formal Concepts,” Chapter 8: “Administration,” and Chapter 10: “Troubleshooting.” The tutorial Chapters 1 through 4 may also be of use, although the setup and installation portion of Chapter 1 will be unnecessary.

The second audience administers an ENVY project or a portion of it. These users might be more experienced developers or nonprogramming system administrators charged with maintaining an ENVY system. These readers will probably find Chapter 5: “Formal Concepts” useful for quickly understanding the ideas behind ENVY, and will definitely want to read Chapter 8: “Administration.”

Finally, there are power users and toolsmiths who understand and use ENVY routinely and want to understand the implementation, advanced features, or ways to customize and extend the system for their needs. For these users the later chapters will be most valuable, particularly Chapter 7: “Extending the System” and Chapter 9: “Goodies.” Chapter 6: “Packaging and Delivery” will be useful for anyone involved in delivering an application, and the tools described in Chapter 8: “Administration” are also valuable for power users and tool builders. Advanced users may also want to skim the tutorial chapters for useful tidbits. Of particular use may be the advice on setting up complex configurations in the “Installation and Setup” portion of Chapter 1.


Conventions

For the most part, the typographical conventions in this book should be straightforward for anyone used to reading Smalltalk code. Code embedded in the text, including the names of classes, methods, and applications, is shown in a different font. For example, we might talk about the method `printOn:` in the class `Question`. The same convention is used for the names of menu items.

We will discuss both Cincom’s VisualWorks and IBM’s VisualAge Smalltalk. Most of what we will say applies to both of these implementations equally, but from time to time you’ll encounter implementation-specific sections. We highlight these as follows.

This paragraph is specific to IBM’s VisualAge. It might be talking about menus in the VisualAge organizer, features that are supported only in VisualAge, or some other aspect of the system that does not apply to VisualWorks.

This paragraph is specific to Cincom’s VisualWorks. It might be talking about parcel integration, features that are supported only in VisualWorks, or some other aspect of the system that does not apply to VisualAge.

One particular stylistic convention for the code deserves note, that of ending methods with a period. The issue of whether to include or omit a closing period is as contentious and emotional as only trivial issues can be. In this book we have opted to include the period, for the following reasons:

- It’s more consistent to always end statements with a terminator (except in the presence of brackets). Some people argue that the period is a separator, not a terminator. In our opinion they spend too much time thinking about parser theory and not enough time about the simplicity of day-to-day work.
- It is less error prone. One of the standard Smalltalk errors is to add a line at the end of a method only to discover that the method will not compile because the previous line has no period. This doesn’t happen if we consistently terminate statements with a period.
- Finally, and perhaps most important, it is more consistent with natural language. One of Smalltalk’s virtues as a computer language is that many of its statements can easily be read as if they were English sentences. I believe the designers of Smalltalk deliberately chose the period because it is the way we end sentences in English.

Software for This Book

This book does not include a CD-ROM containing the software, and this is a deliberate choice. In this era of the Internet, any frozen medium will quickly be out of date. Instead, copies of the accompanying software, along with updates, new goodies, and code contributed by the community will be available at the book’s Web sites.

- [www.us.cambridge.org/titles/0521666503](http://www.us.cambridge.org/titles/0521666503)
- [www.envymasters.com](http://www.envymasters.com)
Links to other software mentioned in the book will be available at these sites. We also list references for the most important outside tools mentioned in the book:

- The SUnit unit testing framework is available at www.xprogramming.com.
- The Refactoring Browser is available at www.refactory.com.
- VA Assist Pro is available at www.smalltalksystems.com.

For Smalltalk software in general, the following links may also be of use:

- IBM’s Smalltalk Web site is at www.ibm.com/software/ad/smalltalk.
- Cincom’s Smalltalk Web site is at www.cincom.com/smalltalk.

VisualWorks and VisualAge are the only Smalltalk implementation currently supported by ENVY but by no means the only Smalltalks available. We list URLs for most of the other Smalltalks so that readers can see some of the other possibilities and breadth of the available implementations, as follows:

- **Dolphin**: A Windows-specific Smalltalk, which is inexpensive, simple to use, and has a very nice user interface. An excellent introduction to Smalltalk is available at www.object-arts.com.

- **Gemstone**: An active object-oriented database that includes its own Smalltalk running on the server. An integration is available that lets you use ENVY on the client to control GemStone code. You can access Gemstone at www.smalltalk.gemstone.com.

- **Pocket Smalltalk**: An open-source Smalltalk for the Palm hand-held computer. It produces extremely compact programs (~45K) and is available at www.pocketsmalltalk.com.

- **Object Studio**: A Smalltalk that comes out of a more 4GL tradition. It is widely used in corporate environments. This Smalltalk is also owned by Cincom and is available at www.cincom.com/objectstudio.

- **Smalltalk MT**: A Smalltalk implementation very tightly integrated with Windows that provides high performance, native Windows threading, DLL creation. It is available at www.objectconnect.com.

- **Smalltalk/X**: Despite the name, a version for both UNIX and Windows NT, with some very sophisticated features. It is available at www.exeftp.de.

- **SmallScript**: While information on this is scarce at the time of publication, this will be a Smalltalk implementation integrated with the Microsoft .NET platform and will be available at www.smallscript.com.

- **Squeak**: An open-source implementation with some very sophisticated graphical features, coordinated by a team at Disney that includes Alan Kay. Squeak boasts a very strong user community and is available at www.squeak.org.

- **Visual Smalltalk**: Once one of the primary commercial Smalltalks, this was abandoned by ParcPlace-Digitalk after an attempted merger. It’s still in use,
but new development is minimal. This Smalltalk was acquired by Cincom as part of the VisualWorks acquisition and is available at www.cincom.com/vse.

As with anything on the web, these addresses are subject to change, so if you have problems with them check the main web site for links or use a search engine.

The Story of This Book

The idea for this book started back in 1998, when Stephane Ducasse of Bern University suggested to Joseph Pelrine that he write a book on ENVY. Soon thereafter, Lothlórien Homet, then struggling to get the SIGS Publications book division back on its feet, sat in on Joseph’s Smalltalk Solutions talk on ENVY and proposed the same thing. Joseph, feeling a bit uncomfortable, agreed on the condition that he could convince one of his colleagues (who knew a lot more about ENVY than he did) to contribute. After some cajoling, Alan Knight agreed to work on the project. While Joseph prepared the book proposal (thanks, Joseph), Alan convinced Adrian Cho of OTI to act as technical editor. Organizing, writing, and many schedule slips ensued. Part way through the project Joseph talked Adrian into coming aboard as an author. After the first draft was written, Alan spent months revising and refactoring. The reason the tome you hold in your hands reads as easily as it does is due in large part to Alan’s work (thanks, Alan). Adrian’s depth of ENVY knowledge, both technical and philosophical, was invaluable and he contributed one of the most useful ENVY utilities to ever see the light of day, the checkpoint mechanism (thanks, Adrian). All this time, Lothlórien Homet has had the patience of a saint, deftly and adroitly juggling carrot and stick to get all of us to finally finish this manuscript (thanks, Lothlórien).

Basel and Ottawa, September, 2000