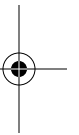
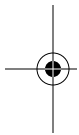


Preface

The aim of this book is to describe a selection of tools and techniques that can be used to script automatic maintenance and configuration tasks. Starting with simple activities and moving to the more complex, we demonstrate how a tiny script running automatically on each workstation can load more substantial scripts from a network server, which in turn can be used to carry out virtually any administration task that you can think of. There are literally thousands of administration tasks that could be automated by scripting. Although we try to give useful examples of some of them, the emphasis of this book is not so much to suggest maintenance and configuration tasks that can be automated; to a large extent, we leave that up to you. Instead, we concentrate on describing tools and techniques that can be put to use in writing scripts that perform any kind of automatic maintenance and configuration tasks that you care to throw at them.

Most of the more advanced scripts we describe involve writing, modifying, and deleting registry keys. Whenever we introduce a key, we try not to simply present it as is, but rather explain what it is, why it lives where it does, and why it is of any importance. Possibly more importantly, we tell you about the tools, techniques, and documentation that we used to find out about some of the more arcane configuration parameters that can be configured usefully with scripts. Our mission is not to mystify by hiding our sources, but rather to equip you to discover for yourself, so you can apply the techniques described here to any workstation-based administrative task that you care to dream up.

This book is about automated maintenance and configuration of NT workstations. Through a series of examples that become more sophisticated and build on one another as the book progresses, it demonstrates the value of scripting for system administrators in countless NT environments. Our emphasis is on technique: we never simply present our scripts “as is”; rather, we walk through the processes and



tools involved in devising, testing, and deploying automated solutions to maintenance and configuration tasks. We hope that by the time you have finished reading the book, you will realize that scripting is an extremely powerful concept that can save enormous amounts of time and greatly reduce the potential for human error in all sorts of situations. Further, we hope that you will feel confident enough to extend some of our examples and develop your own scripts to fulfill the needs of your NT network.

Chapter 1, *Introduction*, gives an overview of the issues involved in configuring and maintaining a fleet of NT workstations. It also introduces the Perl language, the scripting tool used throughout the book.

Chapter 2, *Running a Script Without User Intervention*, discusses various methods of getting scripts (or any other programs for that matter) to run on a workstation without being explicitly invoked by a user. This involves an explanation of a number of registry keys, the creation of accounts for scripting, and use of NT services.

Chapter 3, *Remote Script Management*, develops the techniques discussed in Chapter 2 by showing how to write scripts that always keep themselves up-to-date and introduces the concept of a “stub” script, a sort of mother script that keeps any number of others in check.

Chapter 4, *System Maintenance*, puts these script-invocation techniques to use by demonstrating how many mundane maintenance tasks can be scripted. We give examples of scripts for automating some common maintenance procedures.

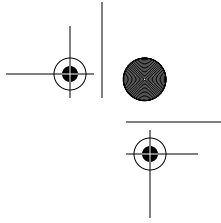
Chapter 5, *Controlling Services and Drivers*, discusses a variety of tools and techniques that can be used to diagnose service and driver problems. This includes an investigation of event logs and some of the more obscure parts of the NT registry.

Chapter 6, *Machine-Specific Scripting*, demonstrates the writing of scripts whose behavior differs depending on the specific workstation on which they are being run.

Chapter 7, *Changing Network Identity*, combines techniques discussed in the previous chapters to illustrate the writing of scripts that can totally change the network identity of a workstation on a non-DHCP network.

Chapter 8, *Script Safety and Security*, takes a step back and discusses the security implications of scripting. It also outlines a development approach that should ensure (as much as possible) the writing of safe, reliable, maintainable scripts.

Chapter 9, *A Custom Module*, takes many script fragments from previous chapters, makes them generic, and shows how they can be wrapped within a single module that, once created, drastically simplifies the scripting process.



The Appendix, *Perl Module Functions*, provides a brief discussion of the use of Perl in the book, as well as a summary of all the module functions that we have used.

Scope and Intended Audience

This book is for any system administrator who has responsibility for managing a network of NT4 workstations and would like to automate maintenance and configuration tasks without resorting to heavy-duty server-based solutions such as SMS. The emphasis of the book is squarely on management of conventional NT4 networks, so none of the scripts we present require the use of Active Directory Services or WMI. Both of these are exciting technologies and are likely to become central to workstation administration in the future, but until they become widely adopted (probably not until after the release of Windows 2000), no one can be totally sure as to the exact role they will play in workstation maintenance and configuration.

It is only fair to warn that some of the specific registry entries discussed in the book, and a few of the tools we present, may not work on Windows 2000 workstations. However, our main emphasis is on techniques and strategies; these will be equally applicable in virtually any workstation environment of the future. In addition, if you adopt the module-based scripting approach discussed in Chapter 9, you should find that any maintenance scripts you write can be adapted with ease if and when you migrate to Windows 2000.

Conventions in This Book

The following typographical conventions are used in this book:

Constant width

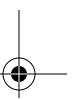
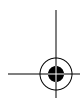
is used to indicate command-line computer output, code examples, registry keys, commands, Perl-language keywords, and keyboard accelerators.

Constant width italic

is used to indicate variables in examples and in registry keys.

Italic

is used to introduce new terms and to indicate URLs, file extensions, filenames, directory or folder names, and UNC pathnames.





This symbol indicates a tip.



This symbol indicates a warning.

We'd Like Your Feedback

The information in this book has been tested and verified to the best of our ability, but mistakes and oversights do occur. Please let us know about errors you may find, as well as your suggestions for future editions, by writing to:

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
800-998-9938 (in the U.S. or Canada)
707-829-0515 (international/local)
707-829-0104 (fax)

You can also send us messages via email. To be put on our mailing list or to request a catalog, send mail to:

nuts@oreilly.com

To ask technical questions or comment on the book, send email to:

bookquestions@oreilly.com

Acknowledgments

Many thanks go to our colleagues in the Computer Centre at Jesus College: Adrian Asher, Rob Feakes, Alison Campbell, and Katie Eagleton. Not only have they given us lots of useful suggestions and made just the right number of sarcastic comments, but they have also kept the place running while we put the finishing touches on the book. Thanks also to Charlie Moore, our network engineer, for accidentally supplying the late-night whisky (noticed it's gone yet?). Many thanks also to Jesus College, for giving us the opportunity to write the book in the first place; in particular we would like to thank the Senior Tutor, Ian DuQuesnay, for

his unfailing support and enthusiasm for anything even vaguely to do with computing.

Robert Denn, our editor at O'Reilly, has been great; his encouragement and suggestions throughout the project have been hugely appreciated. Our three technical editors, Glenn Fincher, Jane Lybecker, and Erik Olson have also helped us greatly with their detailed comments and invaluable suggestions. Thanks also to the rest of the O'Reilly team, who work like mad behind the scenes to make books like this possible. Finally, incalculable thanks go those who we've forgotten to thank, and to the numerous people whose brains we have picked in our time as system administrators.

MML:

In addition to those mentioned above, I would like to thank Martin Keegan and Alistair Turnbull for sharing with me their considerable computing knowledge; thanks also to the librarians at the Royal College of Music in London for sparking my interest in computing in the first place. My Ph.D. supervisor, Dr. Ian Cross, has been very supportive throughout my time at Cambridge, and deserves particular thanks for not nagging me about work while I was writing the book. I am very grateful to the many friends who have not totally forgotten my existence even though my social life virtually disappeared over the last few months, particularly Sebastiaan Kokelaar, Elena Nuttall, and Abbi Wood. Finally, most thanks of all go my parents, Jacqui and Jonathan Lavy, and my two sisters, Stephanie and Philippa, for their unfailing support and patience (!) for many years!

Thanks to my coauthor, Ashley Meggitt, for not sacking me from my day job when I broke the DLT drive!

AJM:

A number of people have supported and encouraged me during the writing of this book, some knowingly, some unknowingly, and some unwittingly. I would particularly like to thank the following: my parents Brenda and Bernard Meggitt who, like "bricks," have always given me more encouragement than can be imagined; my brother, who can "put me down" masterfully, and can only be trumped by gross acts of violence; to my good friend Nick Court who is now too old to beat my half marathon time; to Sandra Last for the continued lunchtime liaisons; to Tim Ritchey for giving me the confidence to write and for pointing out that it was definitely "cool" to write books.

Special thanks must go to my coauthor Matthew Lavy who wouldn't allow me to give up when things went pear-shaped. During the whole writing process he supplied boundless enthusiasm which eventually carried the day. I'm sure that both Adrian Asher and Robert Feakes would also like to join me in thanking Matthew



for breaking the brand new DLT drive, for destroying the spiral binder under the pretext of repairing it, for almost losing a finger through playing with a cable tie, for accidentally rebooting the new server the day after it was commissioned, for . . .

And finally to John Darlington for contributing even less to this book than he did the last!

