

Windows[®] NT

Professional

The essential support source for Windows NT

Using Doskey macros to simplify common Windows NT tasks

By Chris Morton

We recently encountered a situation during a lengthy network configuration and testing operation that required us to repeatedly open a Windows NT command prompt window to ping yet another company host IP address. Since all the IP addresses across our entire organization begin with 144.19, we wished we had a macro utility that we could use to simplify the procedure.

Fortunately, we recalled the Doskey utility that ships with all Windows operating systems. After some experimentation, we realized that this simple command-line utility from the DOS days can still be invaluable, if you remember how to take advantage of it.

In this article, we'll explain how the Doskey command works and show you how to put it to use to save yourself hours of tedious work. We hope this information will shed some light on how this gold mine of a utility can help you solve certain problems you encounter.

What is Doskey?

Basically, Doskey is a small command-line TSR utility designed to help DOS users edit command-line text, recall past commands, search through past commands, execute two or more DOS commands on one command line, and create command macros. In the DOS days, most users took advantage of only the first three features and never really delved into its macro capability. However, its macro capability is what really made Doskey shine in those days. This macro capability can make

Doskey shine again in solving today's problems.

You create a Doskey macro with a simple command that basically assigns a shortcut to a longer command or a series of commands. For example, the command

```
Doskey Dp=Dir /p
```

assigns the command `Dir /p` to a macro named `Dp`. Thus, when you type `Dp`, DOS executes the `Dir /p` command. If you use the command

```
Doskey Dira=A: $T Dir /p
```

you create a macro named `Dira` that represents two commands: `A:`, which changes to drive A, and the `Dir /p` command, which displays a directory listing one page at a time. As you can see, the `$T` character string is used to separate the two commands.

You can also use replaceable parameters in Doskey macros much like you would in batch files. For example, the command

```
Doskey Ty=Type $1
```

creates a Doskey macro called `Ty` that accepts a single parameter. If you then type

```
Ty Myfile.txt
```

Doskey substitutes `Myfile.txt` for `$1` and the command becomes

```
Type Myfile.txt
```

IN THIS ISSUE

1

Using Doskey macros to simplify common Windows NT tasks

3

Understanding how Microsoft networks use the browse service to manage resources

6

Accessing resources that don't appear in your browse list

7

Who wins the master browser election?

8

Creating HTML E-mail templates for Outlook Express

11

Reclaiming valuable hard disk space with CleanSweep Deluxe

14

Disabling the browser capability in Windows NT and Windows 95

15

If you run NT and 95, use the same username for each

16

Professional-strength Windows 98 information

16

Letters

Native American passwords

Windows NT Professional

Windows NT Professional (ISSN 1097-086X) is published monthly by The Cobb Group.

Prices Domestic \$129/yr (\$11.50 each)
Outside US \$149/yr (\$13.50 each)

Phone US toll free (800) 223-8720
Local (502) 493-3300
Customer Relations fax (502) 491-8050
Editorial Department fax (502) 491-3433

Address Please send tips, special requests, and other correspondence to

Windows NT Professional
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

or contact us via Internet E-mail at

ntpro@cobb.com

For subscriptions, fulfillment questions, and requests for group subscriptions, address your letters to

Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

or contact Customer Relations via Internet E-mail at

cobb_customer_relations@zd.com

Postmaster Application to mail at periodicals postage rates is pending in Louisville, KY.

Postmaster: Send address changes to

Windows NT Professional
P.O. Box 35160
Louisville, KY 40232

Copyright © 1998, The Cobb Group, a division of Ziff-Davis, Inc. The Cobb Group and ZD Journals are trademarks of Ziff-Davis, Inc. Windows NT Professional is an independently produced publication of The Cobb Group. All rights reserved. Reproduction in whole or in part in any form or medium without express written permission of Ziff-Davis is prohibited. The Cobb Group reserves the right, with respect to submissions, to revise, republish, and authorize its readers to use the tips submitted for personal and commercial use.

Microsoft, Windows, Windows NT, and MS-DOS are registered trademarks of Microsoft Corporation. All other product names or services identified throughout this journal are registered trademarks or trademarks of their respective companies.

Staff Editor-in-Chief Greg Shultz
Editors Polly Blakemore
Julie Welch
Print Designer Carlo E. Atene
Publications Coordinator Elizabeth Welch
Circulation Manager Amanda Schuster
Product Group Manager Elayne Noltemeyer
Publisher Mark Kimbell

The Cobb Group publishes a full range of journals designed to help you work more efficiently with your software. To subscribe to one or more of these journals, call Customer Relations at (800) 223-8720.

Inside Microsoft Windows 95
Windows 95 Professional
Inside Microsoft Office 95
Inside Microsoft Word
Inside Microsoft Excel
Inside Microsoft PowerPoint
Inside Microsoft Access
Inside Microsoft Office 97

To see a list of our other products, visit our Web site at

www.cobb.com

Note that Doskey \$ variables, while not interchangeable with % variables in batch files, serve the same purpose. To learn more about Doskey options, choose Help from the Start menu, click the Index tab, type *DOSK*, and then press [Enter] twice.

An interactive macro

Now that you have an idea of how Doskey works, let's take a look at how we used it to solve our problem. Initially, we thought we'd create just a single Doskey macro to help us with our ping function. However, we found that we could press Doskey into service for a multitude of tasks.

Our first step was to come up with a shorthand command name for our first Doskey macro. We decided to call it simply *P*, short for *ping*. To create the macro, we opened a Command Prompt window and typed

```
Doskey p=ping 144.19.$1
```

Now when we needed to ping another host PC with an IP address of 144.19.64.9, we could just quickly type *P 64.9* and press [Enter].

Here's how the macro works: During this Windows NT Command Prompt session, Doskey intercepts the *P* macro command and correctly interprets it as ping 144.19. (Note the period after the number 19.) Doskey then parses the additional keystrokes we typed after the *P* command—in this example, 64.9—and substitutes these for the \$1 replaceable variable added to the macro command line above.

In other words, Doskey interprets the six-keystroke *P 64.9* command as if we had tediously typed the full, 16-character PING 144.19.64.9 sequence at the command prompt. Now, you may not think saving ten keystrokes in a single session is a big deal, but over an extended period you'll be amazed at how much time

you save—just in the elimination of typographical errors if nothing else!

Enabling a macro collection

Well, if one Doskey macro is this handy, why not use more? Following the preceding example, let's consider another common TCP/IP administrative function, Route Add. If you really enjoy typing, feel free to repeatedly type *ROUTE ADD 144.19.69.1* into a routing table. We prefer to type *RA 69.1*, however, and let Doskey automatically interpret this shorthand as Route Add 144.19.69.1. Yet another macro could complete the remainder of the entry if you so desired.

This shorthand idea is nice, but having to manually type long Doskey macro sequences in every Windows NT Command Prompt window would defeat any gain in productivity. To eliminate this potential drawback, use Notepad to create a single file that contains all your macros and then save that file in your Windows NT folder.

For example, we created a file called *Dkmacros.txt*, shown in **Figure A**, and added all our macros to it. The syntax to initialize the collection in any Command Prompt window is

```
Doskey /  
macrofile=%windir%\Dkmacros.txt
```

where *%windir%* is a built-in Windows NT system variable representing the folder in which Windows NT is installed.

Final piece to the puzzle

Using a macro collection still involves a lot of typing in a single Command Prompt window. To really be thrifty, you might want to have your Doskey macros pre-loaded, ready to go with in any command prompt session. Fortunately, you can do so by using the */k* switch. Appended after Windows NT's CMD command, the */k* switch runs the command appearing to its right and keeps the Command Prompt window open.

The final trick, then, is to create a Windows Explorer-style shortcut that opens a Command Prompt window and simultaneously loads your Doskey macro file into its environment. To do so, you create a shortcut with the command line

```
CMD /k Doskey /  
macrofile=%windir%\Dkmacros.txt
```

Now whenever you launch your shortcut, your entire collection of custom Doskey macros will be pre-loaded, ready for instant use.

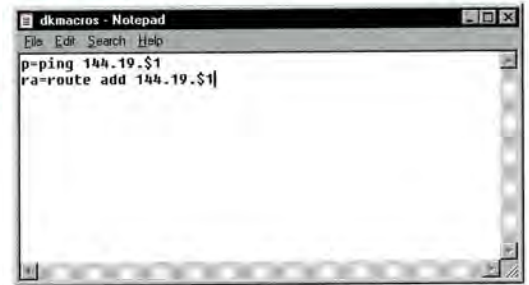
Conclusion

The Doskey command's macro capability is as useful today as it was when it was first introduced. In this article, we ex-

plained how the Doskey command works and demonstrated using it to solve a problem. ❖

Chris Morton is an accomplished Windows author with numerous bylines and seminar presentations to his credit. He has worked with Windows since version 1.03 in 1987. An independent consultant, Chris also provides frequent Windows NT and Windows 95 course instruction to IT professionals through his affiliation with Learning Tree International, Inc.

Figure A



You can create a collection of macros in a single file.