

*Starf now!*

Learn:

*Mastering ir.*

Interactive PowerShell

POMVerShell :

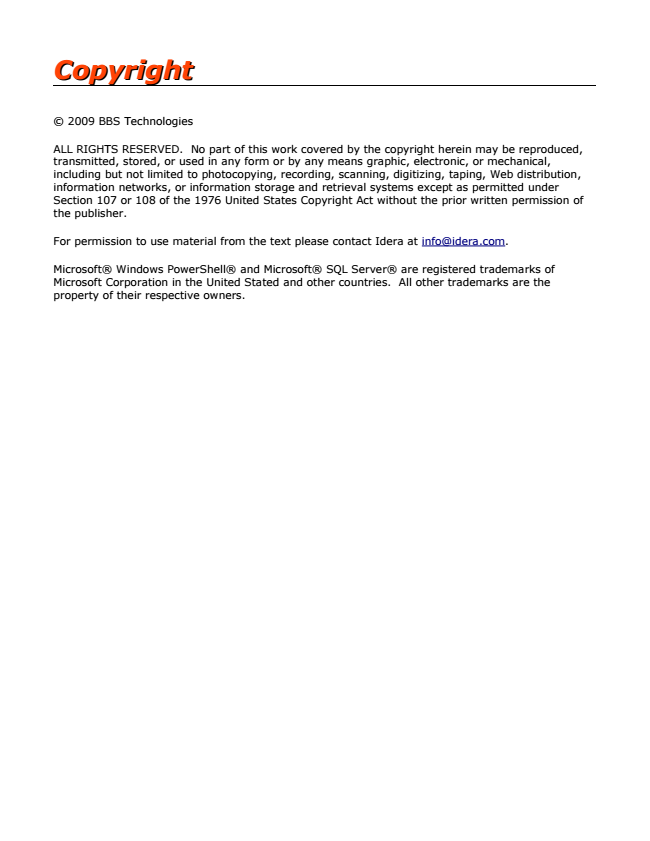
With: Dr. Tobias Weltner, POWerShell MVP

o

|\*|| - - e. \* ". . - - to - 鷺偲鷲 \*鯊醬記g鬣

المها

**N \***

****

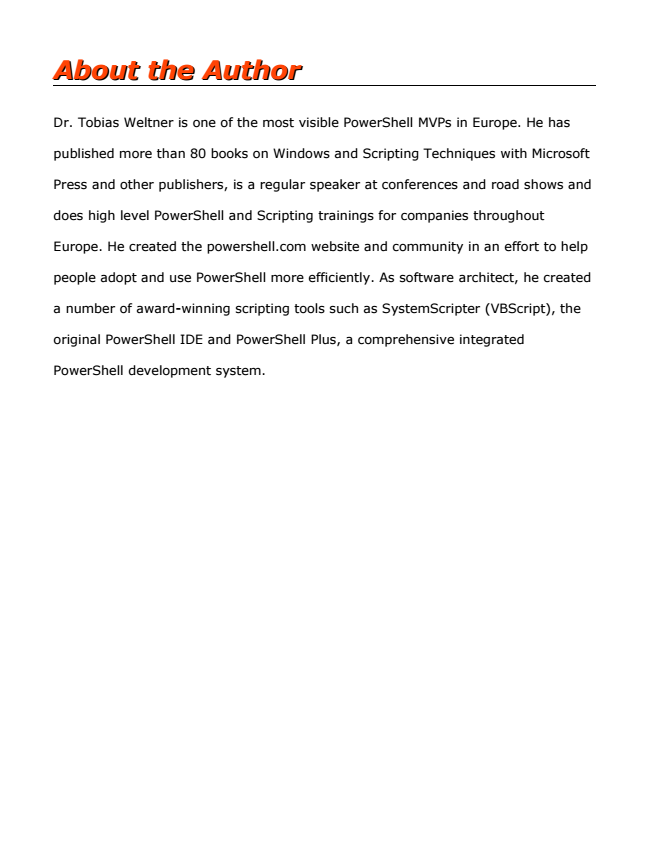
***Copyright Copyright***

© 2009 BBS Technologies

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored, or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems except as permitted under Section 107 or 108 of the 1976 United States Copyright Act without the prior written permission of the publisher.

For permission to use material from the text please contact Idera at info@idera.com.

Microsoft® Windows PowerShell® and Microsoft® SQL Server® are registered trademarks of Microsoft Corporation in the United Stated and other countries. All other trademarks are the property of their respective owners.



***About the Author About the Author***

Dr. Tobias Weltner is one of the most visible PowerShell MVPs in Europe. He has

published more than 80 books on Windows and Scripting Techniques with Microsoft

Press and other publishers, is a regular speaker at conferences and road shows and

does high level PowerShell and Scripting trainings for companies throughout

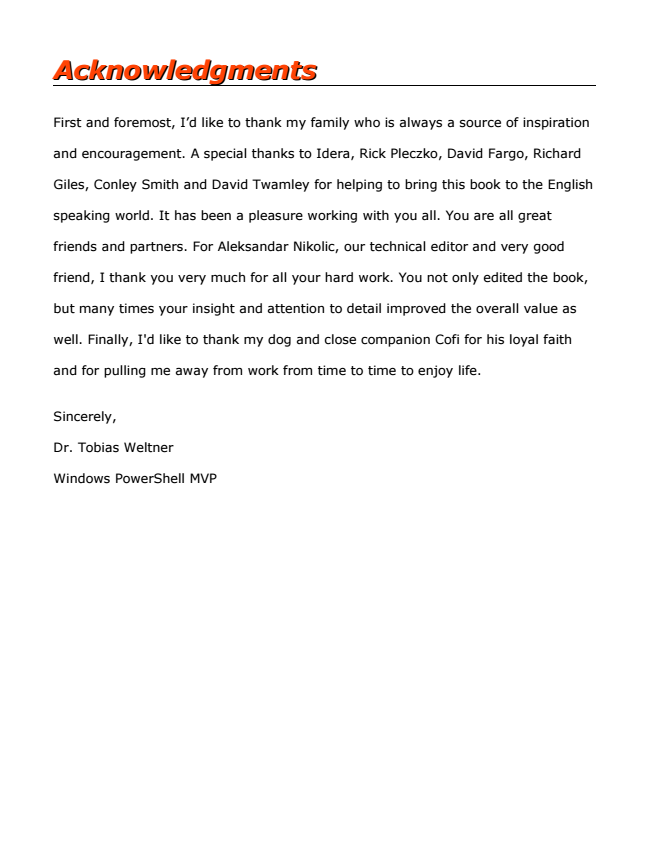
Europe. He created the powershell.com website and community in an effort to help

people adopt and use PowerShell more efficiently. As software architect, he created

a number of award-winning scripting tools such as SystemScripter (VBScript), the

original PowerShell IDE and PowerShell Plus, a comprehensive integrated

PowerShell development system.



***Acknowledgments Acknowledgments***

First and foremost, I’d like to thank my family who is always a source of inspiration

and encouragement. A special thanks to Idera, Rick Pleczko, David Fargo, Richard

Giles, Conley Smith and David Twamley for helping to bring this book to the English

speaking world. It has been a pleasure working with you all. You are all great

friends and partners. For Aleksandar Nikolic, our technical editor and very good

friend, I thank you very much for all your hard work. You not only edited the book,

but many times your insight and attention to detail improved the overall value as

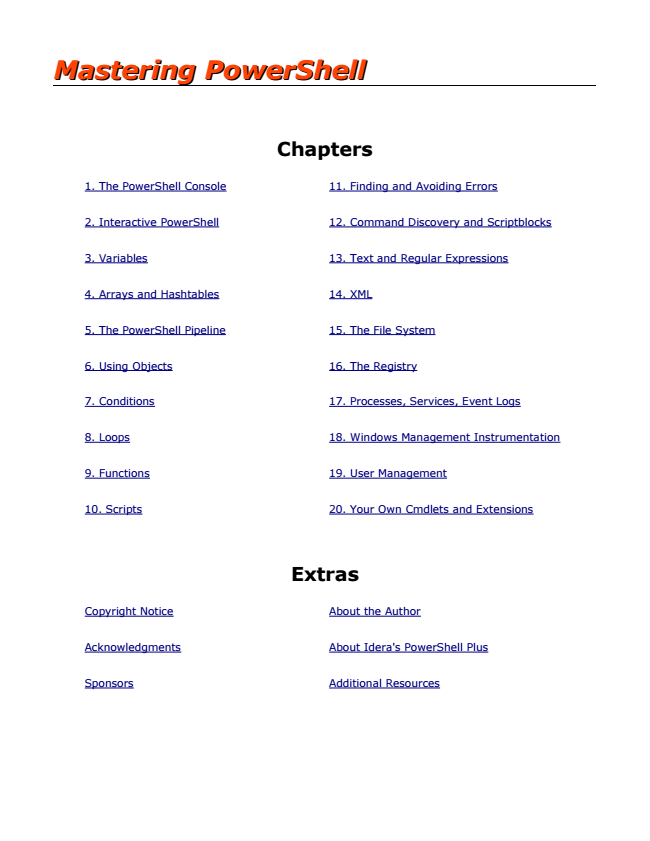
well. Finally, I'd like to thank my dog and close companion Cofi for his loyal faith

and for pulling me away from work from time to time to enjoy life.

Sincerely,

Dr. Tobias Weltner

Windows PowerShell MVP



***Mastering PowerShell Mastering PowerShell***

**Chapters**

1. The PowerShell Console 11. Finding and Avoiding Errors

2. Interactive PowerShell 12. Command Discovery and Scriptblocks

3. Variables 13. Text and Regular Expressions

4. Arrays and Hashtables 14. XML

5. The PowerShell Pipeline 15. The File System

6. Using Objects 16. The Registry

7. Conditions 17. Processes, Services, Event Logs

8. Loops 18. Windows Management Instrumentation

9. Functions 19. User Management

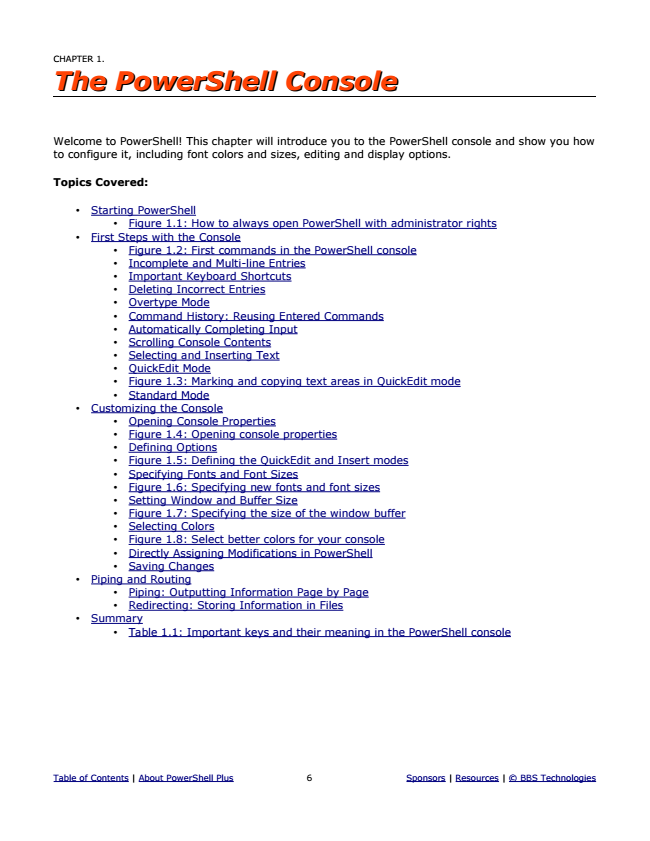
10. Scripts 20. Your Own Cmdlets and Extensions

**Extras**

Copyright Notice About the Author

Acknowledgments About Idera's PowerShell Plus

Sponsors Additional Resources



***CHAPTER 1. The PowerShell Console The PowerShell Console***

Welcome to PowerShell! This chapter will introduce you to the PowerShell console and show you how to configure it, including font colors and sizes, editing and display options.

**Topics Covered:**

• Starting PowerShell

• Figure 1.1: How to always open PowerShell with administrator rights

• First Steps with the Console

• Figure 1.2: First commands in the PowerShell console

• Incomplete and Multi-line Entries

• Important Keyboard Shortcuts

• Deleting Incorrect Entries

• Overtype Mode

• Command History: Reusing Entered Commands

• Automatically Completing Input

• Scrolling Console Contents

• Selecting and Inserting Text

• QuickEdit Mode

• Figure 1.3: Marking and copying text areas in QuickEdit mode

• Standard Mode

• Customizing the Console

• Opening Console Properties

• Figure 1.4: Opening console properties

• Defining Options

• Figure 1.5: Defining the QuickEdit and Insert modes

• Specifying Fonts and Font Sizes

• Figure 1.6: Specifying new fonts and font sizes

• Setting Window and Buffer Size

• Figure 1.7: Specifying the size of the window buffer

• Selecting Colors

• Figure 1.8: Select better colors for your console

• Directly Assigning Modifications in PowerShell

• Saving Changes

• Piping and Routing

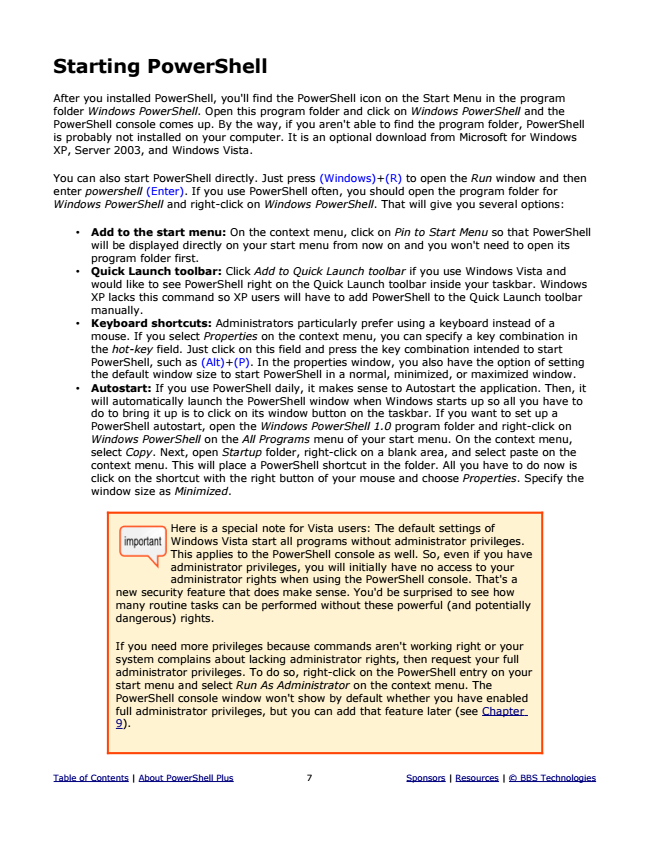
• Piping: Outputting Information Page by Page

• Redirecting: Storing Information in Files

• Summary

• Table 1.1: Important keys and their meaning in the PowerShell console

Table of Contents | About PowerShell Plus 6 Sponsors | Resources | © BBS Technologies



**Starting PowerShell**

After you installed PowerShell, you'll find the PowerShell icon on the Start Menu in the program folder Windows PowerShell. Open this program folder and click on Windows PowerShell and the PowerShell console comes up. By the way, if you aren't able to find the program folder, PowerShell is probably not installed on your computer. It is an optional download from Microsoft for Windows XP, Server 2003, and Windows Vista.

You can also start PowerShell directly. Just press (Windows)+(R) to open the Run window and then enter powershell (Enter). If you use PowerShell often, you should open the program folder for Windows PowerShell and right-click on Windows PowerShell. That will give you several options:

• Add to the start menu: On the context menu, click on Pin to Start Menu so that PowerShell will be displayed directly on your start menu from now on and you won't need to open its program folder first.

• Quick Launch toolbar: Click Add to Quick Launch toolbar if you use Windows Vista and would like to see PowerShell right on the Quick Launch toolbar inside your taskbar. Windows XP lacks this command so XP users will have to add PowerShell to the Quick Launch toolbar manually.

• Keyboard shortcuts: Administrators particularly prefer using a keyboard instead of a mouse. If you select Properties on the context menu, you can specify a key combination in the hot-key field. Just click on this field and press the key combination intended to start PowerShell, such as (Alt)+(P). In the properties window, you also have the option of setting the default window size to start PowerShell in a normal, minimized, or maximized window.

• Autostart: If you use PowerShell daily, it makes sense to Autostart the application. Then, it will automatically launch the PowerShell window when Windows starts up so all you have to do to bring it up is to click on its window button on the taskbar. If you want to set up a PowerShell autostart, open the Windows PowerShell 1.0 program folder and right-click on Windows PowerShell on the All Programs menu of your start menu. On the context menu, select Copy. Next, open Startup folder, right-click on a blank area, and select paste on the context menu. This will place a PowerShell shortcut in the folder. All you have to do now is click on the shortcut with the right button of your mouse and choose Properties. Specify the window size as Minimized.

Here is a special note for Vista users: The default settings of Windows Vista start all programs without administrator privileges. This applies to the PowerShell console as well. So, even if you have administrator privileges, you will initially have no access to your administrator rights when using the PowerShell console. That's a new security feature that does make sense. You'd be surprised to see how many routine tasks can be performed without these powerful (and potentially dangerous) rights.

If you need more privileges because commands aren't working right or your system complains about lacking administrator rights, then request your full administrator privileges. To do so, right-click on the PowerShell entry on your start menu and select Run As Administrator on the context menu. The PowerShell console window won't show by default whether you have enabled full administrator privileges, but you can add that feature later (see Chapter 9).

Table of Contents | About PowerShell Plus 7 Sponsors | Resources | © BBS Technologies

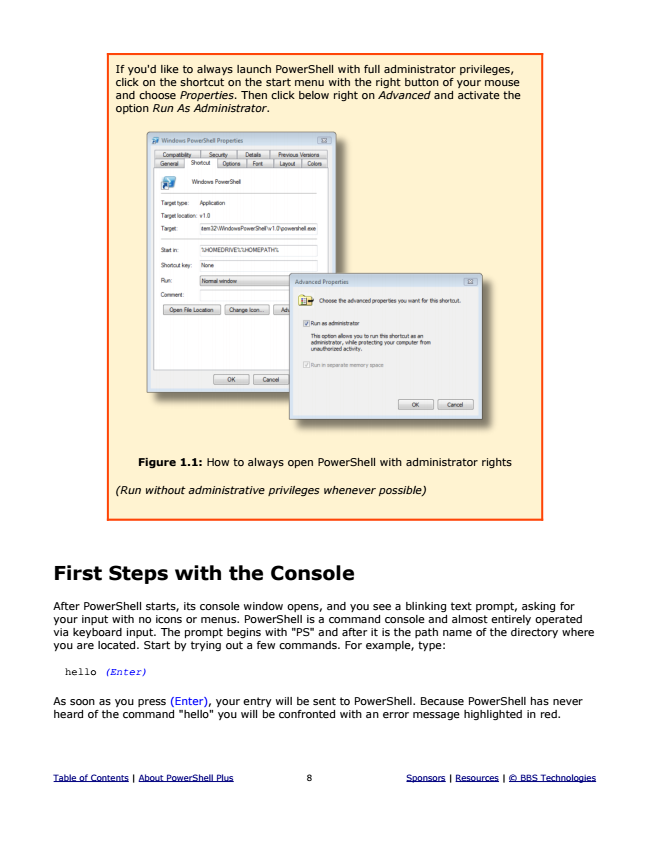


Figure 1.1: How to always open PowerShell with administrator rights

*(Run without administrative privileges whenever possible)*

**First Steps with the Console**

After PowerShell starts, its console window opens, and you see a blinking text prompt, asking for your input with no icons or menus. PowerShell is a command console and almost entirely operated via keyboard input. The prompt begins with "PS" and after it is the path name of the directory where you are located. Start by trying out a few commands. For example, type:

*hello (Enter)*

As soon as you press (Enter), your entry will be sent to PowerShell. Because PowerShell has never heard of the command "hello" you will be confronted with an error message highlighted in red.

Table of Contents | About PowerShell Plus 8 Sponsors | Resources | © BBS Technologies

If you'd like to always launch PowerShell with full administrator privileges, click on the shortcut on the start menu with the right button of your mouse and choose Properties. Then click below right on Advanced and activate the option Run As Administrator.



Figure 1.2: First commands in the PowerShell console

For example, if you'd like to see which files and folders are in your current directory, then type dir (Enter). You'll get a text listing of all the files in the directory. PowerShell's communication with you is always text-based. PowerShell can do much more than display simple directory lists. Just pick a different command as the next one provides a list of all running processes:

*Get-Process (Enter)*

PowerShell's advantage is its tremendous flexibility since it allows you to control and display nearly all the information and operations on your computer. The command cls deletes the contents of the console window and the exit command ends PowerShell.

**Incomplete and Multi-line Entries**

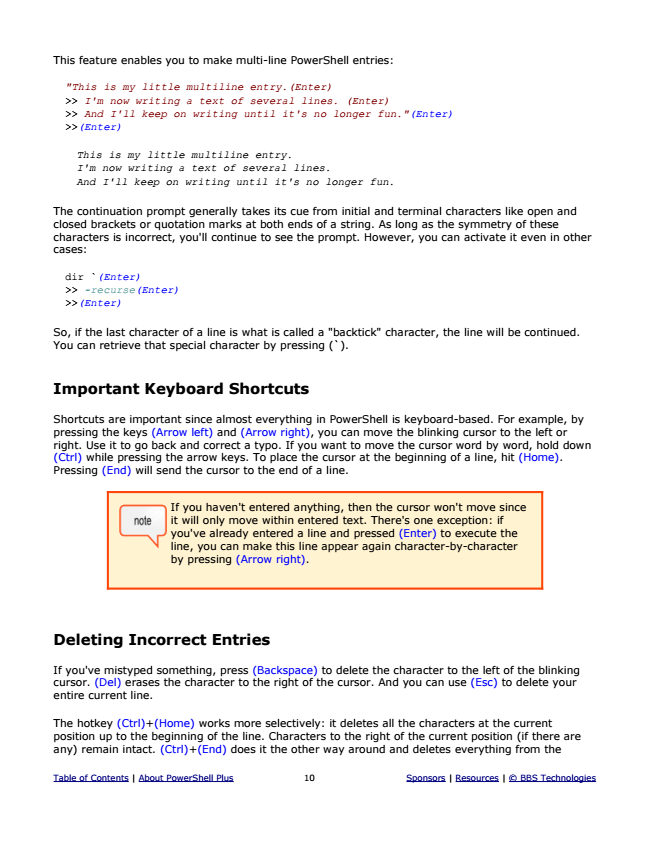
Whenever you enter something PowerShell cannot understand, you get a red error message, explaining what went wrong. However, if you enter something that isn't wrong but incomplete (like a string with one missing closing quote), PowerShell gives you a chance to complete your input. You then see a double-prompt (">>"), and once you completed the line and pressed ENTER twice, PowerShell executes the command. You can also bail out at any time and cancel the current command or input by pressing: (Ctrl)+(C).

The "incomplete input" prompt will also appear when you enter an incomplete arithmetic problem like this one:

*2 + (Enter) >> 6 (Enter) >> (Enter)*

*8*

Table of Contents | About PowerShell Plus 9 Sponsors | Resources | © BBS Technologies



This feature enables you to make multi-line PowerShell entries:

*"This is my little multiline entry.(Enter) >> I'm now writing a text of several lines. (Enter) >> And I'll keep on writing until it's no longer fun."(Enter) >>(Enter)*

*This is my little multiline entry. I'm now writing a text of several lines. And I'll keep on writing until it's no longer fun.*

The continuation prompt generally takes its cue from initial and terminal characters like open and closed brackets or quotation marks at both ends of a string. As long as the symmetry of these characters is incorrect, you'll continue to see the prompt. However, you can activate it even in other cases:

*dir `(Enter) >> -recurse(Enter) >>(Enter)*

So, if the last character of a line is what is called a "backtick" character, the line will be continued. You can retrieve that special character by pressing (`).

**Important Keyboard Shortcuts**

Shortcuts are important since almost everything in PowerShell is keyboard-based. For example, by pressing the keys (Arrow left) and (Arrow right), you can move the blinking cursor to the left or right. Use it to go back and correct a typo. If you want to move the cursor word by word, hold down (Ctrl) while pressing the arrow keys. To place the cursor at the beginning of a line, hit (Home). Pressing (End) will send the cursor to the end of a line.

If you haven't entered anything, then the cursor won't move since it will only move within entered text. There's one exception: if you've already entered a line and pressed (Enter) to execute the line, you can make this line appear again character-by-character by pressing (Arrow right).

**Deleting Incorrect Entries**

If you've mistyped something, press (Backspace) to delete the character to the left of the blinking cursor. (Del) erases the character to the right of the cursor. And you can use (Esc) to delete your entire current line.

The hotkey (Ctrl)+(Home) works more selectively: it deletes all the characters at the current position up to the beginning of the line. Characters to the right of the current position (if there are any) remain intact. (Ctrl)+(End) does it the other way around and deletes everything from the

Table of Contents | About PowerShell Plus 10 Sponsors | Resources | © BBS Technologies