



The IT Support Handbook

A How-To Guide to Providing Effective
Help and Support to IT Users

—
Mike Halsey

Apress®

www.allitebooks.com

The IT Support Handbook

**A How-To Guide to Providing Effective
Help and Support to IT Users**

Mike Halsey

Apress®

The IT Support Handbook

Mike Halsey
Sheffield, UK

ISBN-13 (pbk): 978-1-4842-5132-4
<https://doi.org/10.1007/978-1-4842-5133-1>

ISBN-13 (electronic): 978-1-4842-5133-1

Copyright © 2019 by Mike Halsey

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr

Acquisitions Editor: Joan Murray

Development Editor: Laura Berendson

Coordinating Editor: Nancy Chen

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484251324. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

For my father, James Halsey, who taught me the value of hard work and dedication, and who first introduced me to computers. You set me on the road to my career as an author. I will always be proud of you, and grateful for everything you did.

For Chris Rhodes MVP, MCT. He made everybody feel welcome, valued, and at ease. He will always be missed and never forgotten.

Table of Contents

About the Author	xiii
About the Technical Reviewer	xv
Acknowledgments	xvii
 Part I: IT Support Fundamentals	 1
Chapter 1: An Introduction to IT Support.....	3
The Fundamentals of IT Support.....	5
What?	5
When?	7
How?	7
Never Make Assumptions.....	8
The Language Barrier	8
The Interconnectedness of IT Systems	9
The Human Factor.....	11
Summary.....	12
 Chapter 2: Understanding Your IT System Better	 13
A Brief History of Computers.....	13
What IT Systems Might You Encounter	16
Interface Standards	17
Device Types.....	19
Operating Systems	20
The Interconnectedness of IT Systems	20
Summary.....	22

TABLE OF CONTENTS

Chapter 3: Understanding Your Users: How Much Do They Know? 23

 How to Communicate with Humans..... 24

 Managing Staff Training 26

 Learning Theory 26

 Place Everything in Context..... 29

 Structuring Training and Education..... 30

 Define Your Objectives..... 30

 Mix Things Up a Little 30

 Assess Your Learners’ Knowledge..... 31

 Use Mixed Peer Groups 31

 Help the Learners Consolidate What They Have Learned 32

 Evaluate the Learners..... 33

 Écouter et Répéter..... 33

 Summary..... 33

Part II: IT Support Methodology..... 35

Chapter 4: Flow Logic and Troubleshooting 37

 How Does Flow Logic Work in Troubleshooting?..... 37

 Process of Elimination..... 38

 “Information Is All” 39

 Paperwork Is a Pain, or Is It? 41

 Begin at the End, but Don’t Work Your Way Backward..... 41

 “Don’t Stop Thinking About Tomorrow” 42

 The Impossible Is Possible..... 42

 Work with the Team 43

 Summary..... 44

Chapter 5: Querying Users Effectively 45

 How to Query Users Effectively to Diagnose Problems..... 45

 Users Can Be Anyone, and Anywhere 46

 Never Make Assumptions..... 47

 Yes or No? 47

Take the User with You on the Journey.....	48
The User Is Your Friend ... Yes, Really.....	49
Swipe Left or Swipe Right?	50
The Non-technical Dictionary.....	51
Online Chat.....	51
Summary.....	52
Chapter 6: Joining the Dots: Finding the Root Cause of an IT Issue.....	53
The Beginning of the End.....	55
Working Backward	55
IT Troubleshooting: The Movie	56
The End of the Beginning.....	56
What Are These Dots of Which You Speak?	57
Keeping an Open Mind.....	59
Summary.....	60
Part III: Understanding IT System Problems.....	61
Chapter 7: How IT Systems Are Structured	63
In the Beginning, the Unix-verse Was Created	63
IP Freely.....	65
Aging Tech	66
Windows NT.....	67
Windows vNext.....	69
Creating a New Android.....	70
The Upshot	71
Living in the Internet Age	71
Oh My God! The World Just Ended!	72
The Nearness of You(Tube)	72
Hardware Is Hard Wearing	73
Hardware Also Wears Out	74
Summary.....	74

TABLE OF CONTENTS

Chapter 8: The Human Factor 75

 How the Human Factor and Staff Training Affect IT Systems 75

 Why Users Screw Up IT Systems 76

 Hardware 76

 Software 78

 Settings 79

 IT and Accessibility 79

 Users Are Not IT People..... 80

 The Monkey Mind..... 80

 People Are Complex 81

 Summary..... 82

Chapter 9: The Peripheral Problem..... 83

 Riding the Legacy Wave 83

 Adding Legacy Devices to Windows 84

 Configuring and Troubleshooting Legacy Devices 87

 Troubleshooting Device Drivers 88

 What Else Goes Wrong with Peripherals? 92

 Summary..... 93

Chapter 10: Building and Environmental Factors 95

 The World We Live In 95

 Weather 95

 Sand, Dust, Water, and Moisture 97

 The Built Environment..... 99

 Wi-Fi, Where-Fi Art Thou?..... 99

 Bluetooth and Cellular 100

 Cities and the Countryside 101

 Summary..... 102

Part IV: Documentation and Reporting	103
Chapter 11: Why Good Documentation Matters.....	105
Documentation Saves Time and Money	105
Documentation for Training.....	107
Dumbing Things Down.....	108
Documentation for Troubleshooting	109
Personnel and SLAs.....	110
Getting in Line	110
Engineering Solutions.....	111
Keep It Clear and Concise	112
Summary.....	113
Chapter 12: Creating Troubleshooting Guides	115
Clean, Concise, and Easy to Understand	116
Flow Logic	117
The Dev Problem	119
Now Let Me Tell You a Story	119
The Story Continues	120
What and Why.....	120
So Does the Princess Kiss the Frog?	121
Summary.....	122
Chapter 13: Creating and Managing Paperwork.....	123
First-Line Support Paperwork.....	123
Second- and Third-Line Support Paperwork.....	127
Engineer Paperwork.....	129
Additional Forms and Reports.....	130
Summary.....	131

TABLE OF CONTENTS

Chapter 14: Harnessing System and Error Reporting in Windows 133

Reliability History 133

Administrative Tools 136

 System Information 136

 Performance Monitor 138

 Event Viewer 143

Honorable Mention: Task Manager 148

Summary 150

Chapter 15: Obtaining Advanced Error and Status Information on PCs 151

Getting Detailed Information About Errors 151

 Copying and Saving Event Details 155

 Connect to the Event Log on Another PC 156

Finding Other Windows Error Logs 157

 Text File Logs 157

 XML and ETL Log Files 158

 Dmp Files 158

Summary 159

Part V: Providing Remote Support 161

Chapter 16: Remote Support Tools 163

Remote Desktop 163

Windows Remote Assistance 165

Quick Assist 166

TeamViewer 168

RealVNC 169

LogMeIn 169

Chrome Remote Desktop 170

Summary 171

Chapter 17: Gathering Information Remotely	173
Start with the Asset Tag	173
Permitting Remote Administration of PCs	174
Sign in to the Registry as Another User	177
Using the Microsoft Management Console Remotely	179
Summary	181
Chapter 18: Helping Your Users to Help You	183
Problem Steps Recorder	183
Saving Screenshots	189
Screencasting	190
Xbox Game Bar	190
Summary	191
Index	193

About the Author



Mike Halsey is a Microsoft MVP (Most Valuable Professional) awardee, since 2011, and technical expert. As the author of Windows 7, 8, and 10 troubleshooting books and associated videos, he is well versed in the problems and issues faced by PC users, IT pros, and system administrators when administering and maintaining all aspects of a PC ecosystem. Mike is a teacher and prolific author who uses his training to educate people about complex subjects in simple and straightforward ways.

Originally from the United Kingdom, Mike now lives a simpler and less complicated life in the South of France with his two rescue collies, Evan and Robbie.

About the Technical Reviewer



Massimo Nardone has more than 24 years of experiences in Security, Web/Mobile development, Cloud, and IT Architecture. His true IT passions are Security and Android.

He has been programming and teaching how to program with Android, Perl, PHP, Java, VB, Python, C/C++, and MySQL for more than 20 years.

He holds a Master of Science in Computing Science from the University of Salerno, Italy.

He has worked as a Project Manager, Software Engineer, Research Engineer, Chief Security Architect, Information Security Manager, PCI/SCADA Auditor, and Senior Lead IT Security/Cloud/SCADA Architect for many years.

His technical skills include Security, Android, Cloud, Java, MySQL, Drupal, Cobol, Perl, Web and Mobile development, MongoDB, D3, Joomla, Couchbase, C/C++, WebGL, Python, Pro Rails, Django CMS, Jekyll, Scratch, and so on.

He worked as visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University). He holds four international patents (PKI, SIP, SAML, and Proxy areas).

He currently works as Chief Information Security Officer (CISO) for Cargotec Oyj, and he is member of ISACA Finland Chapter Board.

Massimo has been reviewing more than 45 IT books for different publishing companies, and he is the coauthor of *Pro Android Games* (Apress, 2015), *Pro JPA 2 in Java EE 8* (Apress 2018), and *Beginning EJB in Java EE 8* (Apress, 2018).

Acknowledgments

With many thanks to Gwenan Spearing and James DeWolf for making all of this possible.

PART I

IT Support Fundamentals

CHAPTER 1

An Introduction to IT Support

There's often an interesting story as how people got started with a career in IT support. In my case I was a tinkerer; I wanted to know what was inside the case and how things were made. This meant that whenever I had access to a computer, be it my own Sinclair ZX81, Spectrum or QL, or the Apple II or IBM PC that my father brought home from work, I would pull it apart to see what it was made of.

It's possible really that I could have been an engineer though I didn't have much of an understanding of semiconductors or electrical resistance. Studying electronics for a while when I was 16 didn't really help either as I was far more interested in programming and the user experience.

Inevitably however this led me to build an aptitude with computers which my parents spotted early on and encouraged. From the age of 11, I was never very far from a computer, even having one as my constant travel companion for most of my late teens and right through my 20s in the form of the Psion Organisers and Series 3 and Series 5 PDAs. Today I own the reborn Psion, the Gemini, for which I demonstrated my love for the form factor by becoming the person who wrote the official user guide.

When you have such constant and close experience using computers, it's easy to build a relationship with them where you understand how they function, what makes them operate in different ways, and what's hidden away beneath the surface.

This of course is where the story gets interesting and perhaps a little funny. I still tinkered in my 20s, only now I was tinkering with software and operating systems. Playing with .ini files in the early versions of Windows, or boot partitions and registry entries. It wasn't long before I would regularly begin to break my PC. This wasn't a problem at the time as I wasn't using it for work, or anything critical, and had the time to teach myself how to diagnose what I'd done, and ultimately how to fix things.

When I discovered a tweak or a hack that was particularly cool though I wanted to share it, and so would implement it on the computers of my friends and family; they needed me to provide tech support and because they didn't understand the mechanics of what I was doing rarely questioned things.

You can probably tell where this led, and pretty soon I was not just breaking my own computer but theirs as well. This was slightly more of a problem as they'd be annoyed. I would have to fix things, quickly, efficiently, and effectively, and it's amazing how quickly you can learn how to repair problems when somebody's breathing down your neck waiting to get access to their email again.

It was at that point that I began to do IT support for a living, first independently helping individuals with PC problems in their homes, and then for Fujitsu Siemens as it was then in second-line support in a call center. It was my time providing support for major banks, supermarkets, research firms, and retail giants that taught me just how some people could mistreat their PCs, hardware and software, and cause endless problems.

One particular story that always raises a laugh with me is a colleague who took a call from a manager who had decided that his keyboard was dirty and needed cleaning. He'd filled his basin with hot, soapy water and given the keys a good scrub. Recognizing though that it was an electrical equipment, he'd hung it upside down overnight to give it time to dry.

The following day his computer wouldn't work, so he called the IT helpdesk and explained what he'd done. On checking the asset tag information the manager had provided, my colleague had to inform him that the reason his computer wouldn't work was because the keyboard he'd washed was built into the rest of his laptop.

My colleague was as you would expect a consummate professional, and only laughed his head off and told the rest of us what had happened, after he'd arranged for the laptop to be replaced (there wasn't a lot of point in servicing it), ended the call and written up his notes.

There are definite protocols to follow when providing IT support, and openly laughing at the customer rarely sets the right tone, no matter how funny or idiotic the situation they found themselves in might be. We've all heard the story of the person who couldn't get their computer to work, but who couldn't see around the back of the unit to check the power and monitor cables as the lights were off because of a power cut. We've also all heard the story of the server technician who was complaining his keyboard had packed up, only to eventually find another keyboard sitting underneath it that worked perfectly.

One of my favourite stories doesn't involve IT support at all, but rather a PC retail outlet, a large chain, which a friend was visiting one sleepy Sunday with his father. He called me to say that the sales guy was following them around the shop floor, and asked what he should do; on that occasion I have to admit I did laugh.

The Fundamentals of IT Support

There are many roles in which you might find yourself providing IT support, from first-, second-, or third-line technical support, on-site or traveling engineer, systems administrator or the manager of a team of administrators, the owner of a small store or business that repairs PCs for customers, or someone with an aptitude for computers who repairs problems for friends and family.

All IT support however stems from three fundamental questions. What, when, and how? What is it that's changed or that happened just before the problem began? When did the problem begin? How did the problem begin?

This last question is actually the most important as the core desire of anybody providing IT support is to reduce their own workload and stop other people from being a numpty.¹ If you can configure their computer in a way as to prevent that problem from recurring, or help the user understand what they did so as to ensure they don't do it again, then that's less time you'll spend slapping your hand against your face, and more time you can devote to playing World of Warcraft. So let's look in more detail at these three questions, as they're going to be something I'll mention a lot.

What?

The question "What?" is the most basic principle of IT support, and it's utterly impossible to provide any kind of support without it being asked. It's slightly more complex however than "What the hell have you done now?" or "What could possibly have convinced you that was a good idea?"

I always start with the question "What's changed?" as nothing ever goes wrong with computers unless something has changed. They always work out of the box which is why it's often said that a computer that's left inside the box, and never used, will never

¹Numpty ['nəm(p)tē]: *Noun* (British informal), a stupid or ineffectual person who has little idea what they're doing or talking about.

develop a problem. If you can understand what it is that's changed, or that has happened recently, then you can often get to the root of the problem very quickly.

Let's look at some scenarios, because as you might have already guessed by now, I'm quite fond of those.

Scenario A A person is complaining they can no longer print to their printer. On asking the question what's changed, it transpires that the printer developed a fault and was swapped for a new one a couple of days before. Externally, and to the untrained and, let's face it, uninterested eye of the office worker, the new printer is completely identical to the old one, except that the new printer has an added "S" on the end of the model number, a tiny change that can have all sorts of ramifications for drivers, default printer setup, and tray selections.

Scenario B A worker cannot get access to cloud storage so they can open documents they need for a project. On asking the question "What's changed?" you might discover that all the PCs in the office installed some Windows Update the evening before as people left for the day and that three of this individual's colleagues have retired to the kitchen for a cup of tea as they can't access the remote files either.

Scenario C A remote worker can't get access to the company network to upload their sales data, but hasn't contacted their workplace directly as this is what IT support is for. A quick call to the workplace, or a look online at the ISP's (Internet Service Provider) web site, reveals that somebody in a digger has accidentally severed the main broadband fiber connection while working on the construction site up the road.

If you understand what it is that has changed, you can narrow down the number of possible causes for the problem. This is what I like to call the Sherlock Holmes method, and indeed the "world's greatest detective" probably would have been very good at IT support.

Sherlock Holmes, or rather the author Arthur Conan Doyle, stated that "Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth." Turning IT support into a process of elimination is essential as there are just so many things that can go wrong. We'll look at these in more detail later in this chapter.

When?

In order to understand what a problem is, and the possible knock on effects and ramifications it can have, you need to know when it began. It might be that the problem occurred as people arrived for work that morning, as in scenario B. Alternatively it could be that the problem has existed, on and off, for several weeks. Julie first encountered it in accounts, and Dave in logistics had it too a few days later. It's been on the caller's PC now for some time, but because they don't use the app/feature/hardware on which there is a problem, they've not thought too much about it until now.

Tracing problems back that began some time ago can cause problems, and this is where you can use features on PCs such as the Event Viewer and Reliability Monitor, both of which we'll look at in Chapter 14. You might discover however that the problem occurred just after all the desks were moved after the annual spring clean, or around the same time as a massive thunderstorm. All of this is useful information that helps you narrow down the possible causes.

How?

This leads us onto "How?" the problem occurred, but even this is more complex than it might at first appear. The problem occurred when I turned off my PC. Okay, but how did you turn it off? Did you use shutdown from the Start Menu, press and hold the power button for 4 seconds, or just switch it off at the mains socket?

In another example someone might have a problem with a tablet that happened because a software update was installed. In fact on this occasion it could be pure coincidence that the software update occurred around the same time as the problem began, and the actual cause of the problem is a change to security policies requiring a certificate import on their device they didn't read the email for because they've just returned from vacation.

People don't want to know technical things, they see computers as consumer electronic devices in much the same way they view their TV or microwave. This isn't helped by the fact that their TV might occasionally get a software update, or the PC is a tablet with an embedded OS (operating system) and apps that just come from a store.

This means that asking the question "How?" might just return a puzzled look and the response "You're the IT person, you tell me." On these occasions asking how probably won't get you very far, but you can usually ascertain the information you need from having asked what is it that has changed.

Never Make Assumptions

Patience is a virtue; in IT ... doubly so. Okay, that's not actually a quote but it's a good rule of thumb. You should never make assumptions about people, circumstances, hardware, software, and apps, or problems as in doing so you'll be limiting your diagnostic abilities and you're very likely to come to incorrect conclusions.

You can't assume a cloud service might have an outage when it transpires the user has connected their laptop to the wrong Wi-Fi network. You also can't assume a printer driver is misconfigured and needs reinstalling when it turns out the user is actually visiting a different office on that day. Nor can you assume the user simply doesn't know what it is they're doing, when it really transpires a new version of the software they're using has just been deployed, and half the features they use every day have moved or changed in some way.

You also can't make assumptions about what people know about computers and technology. Not everybody is as technically literate as you are, not everybody has used the same technology and software you use as a matter of course, and not everybody knows the difference between a USB port and an SD Card reader. This of course brings us neatly on to ...

The Language Barrier

I'm pretty sure you've all encountered someone pointing at their monitor and referring to it as the PC, when in fact the PC is a large black box that sits under their desk. You might also have encountered someone holding up their Surface Pro and referring to as an iPad, or as in the case of my own household just two days before I wrote this, turning the mesh router system off and on again to fix an Internet connection problem, when it was actually a separate modem that connected the house to the World Wide Web.

It's common for non-technical people to say that "The Internet isn't working" when in fact they mean it's one web site they're having trouble with, and only because they've forgotten their password or left their phone at home and now the two-factor authentication doesn't work.

Language is hugely important here in helping ascertain what it is that has happened. You might be perfectly familiar with terms such as UEFI, store app, 256-bit AES, USB-C, or Developer mode, but the average person is likely to look at you funny if you start referring to these things over dinner.

We also live in an increasingly small world, where people from one country work and live in many others, where traveling across the planet for work is increasingly common, and where conference calls between workplaces in different timezones is just a perfectly normal Friday afternoon.

Cultural and language differences can have a huge impact on identifying what the cause of a problem is. Somebody for whom their native language is the only one they speak is unlikely to be able to describe the paper coming out of the printer with nasty black lines across it, and somebody in a country where use of your language is different (such as the way people in countries like India and Pakistan use English) can mean it's sometimes difficult to ascertain the exact meaning in what people say.

The Interconnectedness of IT Systems

This brings us back to Sherlock Holmes and IT support being a process of elimination. You've asked the questions What, When, and How, and hopefully you're already closer to identifying the problem, its cause, and the solution for it. Sometimes though the problem can be enormously complex and have so many possible strands that you might wonder how you'll solve it at all.

Let's look at a scenario in which workers have just moved into a newly refurbished office, only to find the network connection is intermittent and keeps failing. At this point everything is open to being a possible cause, and you need to ask more questions to get to the root of the problem.

Is the problem affecting every computer, or do some appear exempt? If the response is that Jeff using his BYOD (Bring Your Own Device) iPad seems to be fine, what's different about his situation? It could transpire that he's connected to a different Wi-Fi network, one set up specifically for BYOD devices that doesn't include access to critical file shares.

This is useful information as we now know that it's probably **not** an issue with the Wi-Fi network(s) or the routers as he'll be connecting through the same hardware. This though still leaves us with several possibilities ...

Could it be the file server in some way, perhaps with security or permissions settings (unlikely given the circumstances, unless more users are connecting than have been allowed in the server configuration). Could it be the switch hardware the server is connected to, or perhaps even the server hardware itself? Could it be that the server is in the cloud and it's an intermittent Internet connection issue that the ISP is already aware of?

Could it be malware that's crept across the company network and is using PCs to conduct as massive Distributed Denial of Service (DDOS) attack on a major corporation, and that as Jeff is using an iPad it hasn't been infected?

It's here that we begin to grasp the complexity of modern PC systems and how interconnected everything is with everything else. Our computers are "always-on" devices, which means they are almost never without a connection to the Internet. This means that a third-party app update service running as a process on your PC is connected to your Wi-Fi network, then to your routers and switch hardware, on to cabling that runs through the building and across town to the telephone exchange, fiber, and satellite connections that connect the exchange to your ISP, more fiber, and satellite connections that link your ISP to the national infrastructure and on to other countries, in one of which sits a server farm containing a copy of your files and documents. That server runs in a virtual machine (VM), running as one of a dozen VMs on one of a thousand rack servers. All of these things are connected together, because they're all making a connection to each other and communicating pretty much all the time.

As much as this interconnectedness makes computer systems complex on a macro-level, on a micro-level the complexity doesn't go away. We have multiple apps and drivers, which will likely never have been tested long term with one another due to the almost limitless combinations of software and hardware that exist in the world, running on circuit boards and silicon that can contain components as small as 5nm (nanometers) across.

The upshot of all this interconnectedness is that every computer on the planet is theoretically, simultaneously connected to every other computer on the planet. This is how we can instantly access servers in the arctic circle, while simultaneously getting remote access of a PC running a completely different operating system on another continent, download email and receive instant messages from people on the far side of the world, and make a video call to someone elsewhere in our own building.

This is a conceit really, as firewalls, security policies, and structural limitations don't actually mean your Dell laptop is connected to the PC of the finance minister in China. I use it though to highlight how complex our networks can be, and how many devices – PCs, laptops, tablets, smartphones, printers, NAS (Network Attached Storage) drives, servers, datacenters, fitness trackers, PDAs (Personal Digital Assistant), games consoles, televisions, smart speakers, fridge freezers, cars, robots, security cameras, door locks, and Terminators – we're theoretically making a connection to at any one time and of all the things that could possibly be contributing to the problem you face; okay so I'm exaggerating about being connected to Terminators ... or am I?

The Human Factor

Earlier in this chapter, I covered some of the difficulties you can face dealing with people. Computers are much easier to work with; they all speak the same language, or can at least be configured easily to speak ours, they're reliable and always operate in the same way, and they're always consistent in the messages they give us.

Sadly, human beings are infallible and infinitely complex. What makes complete sense to one person will make absolutely no sense to many others, and one person being able to accurately and eloquently describe a problem might be completely different to another person being unable to find the words, or the correct terminology when you need them too.

All of this needs to be taken into account when speaking with someone about an IT problem and I approach things using the following steps.

1. Assume nothing except that the person you are talking to might have no technical knowledge at all, and might not even speak the same language as yourself.
2. If 1 proves to be false, adapt quickly to bring your level of questioning to their level; do not patronize them or insult their intelligence by asking if they're sure they're holding the mouse the right way up.
3. Ask questions to which the answers will be short, preferably just Yes or No; this will help reduce misunderstandings.
4. Work through a checklist of common problems and solutions; we'll look at this in much more detail in Chapter 13. This can help eliminate problems that are simple to fix. We might all hate first-line support call centers, but they serve an important function.
5. Do not speak too quickly, take the user with you on the journey. Remember that you only want to speak to them about this problem *once*.
6. If you ask the user to perform checks or actions, ask them to repeat what you've told them before they do. This checks they understand, can perform the task(s) correctly, and helps them consolidate those actions should the error recur.

7. Don't give the user too many things to do at one time, or instructions that are too long; you'll simply lose them.

Following these steps can help avoid confusion, diagnose problems more quickly, and effect solutions more, well ... effectively. Fortunately there are also things you can do with human beings that can really help and assist you in the diagnostic process, no ... really! In Chapter 16, I will detail tools you can use to enable a user to demonstrate what they were doing at the time the problem occurred, and in Chapter 14, I'll show you how to use the Event Reporting system to alert the user whenever an error or problem occurs they may not be immediately aware of.

Most of the information you need to diagnose problems on computers though can be obtained from the computer itself, and this is something I'll detail in Chapters 7 and 8.

Sadly it's true that other operating systems, such as Google Chrome and Android, and iOS don't provide this information in any way near the same level of detail as can be found in Microsoft Windows, though where this is available it will be mentioned and discussed.

Summary

It's clear that IT support is a process of elimination, a communication challenge as well as a technical one, and that the days of a computer being a standalone device, not connected to anything other than a keyboard, mouse, and printer, are long behind us. This can make the process of providing IT support more challenging but can also open opportunities for providing better support.

In the next chapter, we'll look in much more depth at what we've covered here, covering subjects such as flow logic, effective questioning of users, and how you can join the dots to build a complete picture of the problems you face.

CHAPTER 2

Understanding Your IT System Better

What is a computer? If you asked 20 different people what a computer was, there's a good chance that you'd get 20 different answers. Some might say it's the box on my desk at work, while others might point at the screen and say it's that. Some people will say it's their iPad, when they actually have an Android tablet, and other people who own a Chromebook might say it's their laptop.

It's very true then that the definition of what a computer is is very different to what we might have said 20 years ago.

A Brief History of Computers

I love potted histories, and the history of computers is always an exceptional story. The definition of the word computer at [dictionary.com](https://www.dictionary.com) is as follows:

A programmable electronic device designed to accept data, perform prescribed mathematical and logical operations at high speed, and display the results of these operations.

Back in the days of the first computers, it was very logical and only applied to a very small number of machines. However these days computers are everywhere; we have them in our cars, in our kitchens, and in our televisions; and I'll talk about the interconnectedness of IT systems shortly.

The first ever electric programmable computer was created in the United Kingdom in 1943 to help British wartime code-breakers to decipher encrypted German messages. Colossus doesn't exist any more in its original form but was rebuilt in recent years by experts at the National Museum of Computing at Bletchley Park (UK); see [Figure 2-1](#).

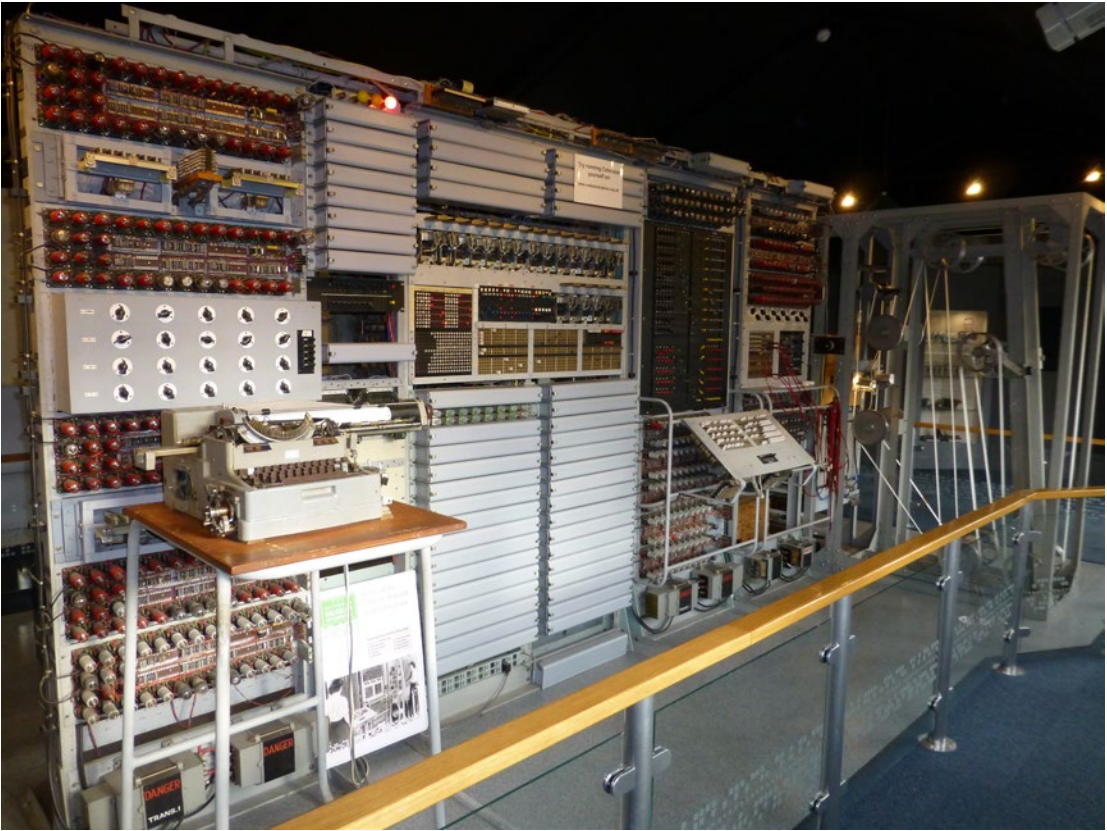


Figure 2-1. *The Rebuilt Colossus Computer at the National Museum of Computing, Bletchley (UK)*

For many years people considered a computer to be the large beige box that sat on their desks, and took up far too much space with its attached cathode ray tube (CRT) monitor. For many years I owned, and loved, an Olivetti M240 PC with green monochrome screen, and lowly 8086 processor; see Figure 2-2.



Figure 2-2. *The Olivetti M240 PC*

Nowadays people would often consider an iPad, PC, or a Microsoft Surface Pro to be a PC. Indeed they have the power to perform daily computing tasks (though you still need a full desktop system for intensive tasks such as gaming and video production and rendering).

Indeed our smartphones have the same computing power as the top-end desktop PCs from only a few years ago, and some can be used with a keyboard, mouse, and monitor to give you a full PC experience.

I myself use the Gemini and Cosmo PDAs from UK tech startup Planet Computers; see Figure 2-3. The Gemini and Cosmo are reborn upgrades to the legendary Psion Series 3 and Series 5 PDAs from the 1990s and also to the hugely popular Nokia Communicator, and although they might have a small screen and keyboard, they're completely pocketable, eminently usable, and capable of performing a wide variety of basic and complex computing tasks, on your choice of Android, Linux, or Sailfish OS. Additionally, they enjoy smartphone levels of battery life, so beat laptops in that regard as well.



Figure 2-3. *The Planet Computers’ Gemini PDA*

It is interesting to note though the very first computers all came about because of a need from governments and the military during the Second World War. A great many of the technological advancements and products we use day to day came about from military or academic research.

The World Wide Web (later known as the Internet) was originally a way for universities to share knowledge, but the technology can be traced back to the The Advanced Research Projects Agency Network (ARPANET), a computer network devised by the US military during the Cold War.

Where your technology will lead us is anybody’s guess. As I write this, foldable screen devices are being prototyped by major tech companies, and are being prepped to become the “next big thing.” How successful they’ll be is anybody’s guess at this stage, though when you read this they might already be old hat.

To think that the first computers were invented only 70 years before this is incredible, and the advances that have been made in silicon manufacture and software can only make looking to the future even more exciting.

What IT Systems Might You Encounter

Now that we’ve established that a computer could be literally anything from a desktop to a PDA, and might be found anywhere from an office to the inside of a car, we should look at the different types of hardware and operating system you’ll be facing.

If you provide IT support for a large company, this job will be slightly more straightforward for you (though there is a caveat with this) in that every device will have

an asset tag, and that tag will identify exactly what the device is, who manufactured it, how old it is, and so on.

The caveat of course being that many businesses these days encourage the use of Bring Your Own Device (BYOD) computers (often known in IT circles as Bring Your Own Disaster) as a way to both cut costs and have people using computers that they're personally happier and more comfortable with, but mostly to cut costs.

So what would you be facing with IT systems that might not be tagged? Well this is much more complicated than just saying it'll be a PC with a monitor, keyboard, mouse, and printer. Let me explain.

Interface Standards

Business users of computers are, generally speaking, a pain in the butt in their refusal to ever spend money on the one thing that they rely on to keep them operating, their IT equipment. This means it's very common to find a payroll system using an old dot-matrix printer, or an engineer who relies on a serial device for taking measurements. So what different interface standards might you encounter, and why are people still using them?

USB

The obvious interface standard to begin with is USB, but don't be fooled into thinking that there's one USB standard to rule them all, as there just isn't.

For starters, there are six different types of USB plug: Type A which is the rectangular one we always plug in the wrong way up; Type B, which is the most commonly used for printers; Mini and Micro USB that are used for everything from microphones to smartphones; Micro-B that you'll often see on external hard disks; USB-C which is now becoming far more common; and even variants such as Apple's Lightning connector.



Figure 2-4. *Different types of USB plug*

These can then come in different speed and bandwidth variations including USB (known as USB v1), USB 2, USB 3, USB 3.1, and USB 3.1 Thunderbolt. Don't confuse USB and USB-C in this regard either, as if somebody tells you they need a Thunderbolt cable, is it then a USB Type A, or a USB Type C cable they're after?

Firewire

Firewire is not very uncommon and was replaced by USB after just a few years. It's another plug and play technology though like USB that was largely used for video and musical equipment, so you might still find it used in some specialist circumstances.



Figure 2-5. *Firewire plugs have different varieties, like USB does*

Serial

Serial devices, where you typically have a 9-pin male to female plug/socket arrangement, are still very popular in specialist industries such as engineering because they allow for analogue data transfer. Serial plugs are capable of transmitting and receiving data at different baud rates (the modulation rate of an electrical signal). This means they are often used in dedicated monitoring roles.

Serial devices have to be manually configured as “Legacy” devices on modern computers, as per instructions provided by the hardware manufacturer for the device. This is why engineers tend to get very excited when they see second-hand laptops on sale that have a Serial port built-in.



Figure 2-6. *Serial plugs*

Parallel

If you find an old dot-matrix printer hidden in the corner of an office, but still in use, you can bet it will be a Parallel device. This is a fork of the Serial interface that allows multiple data channels to broadcast at once (serial only allows for one). They were almost always used for printers, and they need to be configured as “Legacy” devices on modern computers.



Figure 2-7. *Parallel plugs*

Device Types

This brings me to the different devices you’ll find using these plugs. A standard computer system will have the main PC, a monitor, keyboard, mouse, and printer, but it doesn’t stop there.

Other devices you will encounter include, but are not limited to, games or other specialist controllers and professional keyboard setups, microphones, scanners of various types, accessibility input systems such as braille interfaces, musical and video equipment, external storage devices, and the list goes on.

Operating Systems

Lastly there are all the different operating systems you'll encounter. Windows XP has been out of support for a number of years now, but that doesn't mean you won't still find people and companies using it. Windows 7 is out of support in January 2020, and Windows 8.1 is out of support in January 2023.

If people are using Windows 10, which version are they using? Each "feature pack" for Windows 10, the major updates that are released twice a year, and only officially supported for a couple of years, unless a PC is on the Long Term Servicing Channel, where they get support for up to 10 years.

Apple only supports OS X and iOS versions for a couple of years, and Google does the same with versions of Android. This last one can present a problem as there are a great many mobile devices out in the world running long-expired and completely unpatched versions of Android.

Linux has a similar life-cycle these days too, so it's essential to know what operating system you're facing and what the support and security implications of that might be.

The Interconnectedness of IT Systems

This brings us back to Sherlock Holmes and IT support being a process of elimination. You've asked the questions What, When, and How, and hopefully you're already closer to identifying the problem, its cause, and the solution for it. Sometimes though the problem can be enormously complex and have so many possible strands that you might wonder how you'll solve it at all.

Let's look at a scenario in which workers have just moved into a newly refurbished office, only to find the network connection is intermittent and keeps failing. At this point everything is open to being a possible cause, and you need to ask more questions to get to the root of the problem.

Is the problem affecting every computer, or do some appear exempt? If the response is that Jeff using his BYOD (Bring Your Own Device) iPad seems to be fine, what's different about his situation? It could transpire that he's connected to a different Wi-Fi network, one set up specifically for BYOD devices that doesn't include access to critical file shares.

This is useful information as we now know that it's probably **not** an issue with the Wi-Fi network(s) or the routers as he'll be connecting through the same hardware. This though still leaves us with several possibilities ...

Could it be the file server in some way, perhaps with security or permissions settings (unlikely given the circumstances, unless more users are connecting than have been allowed in the server configuration). Could it be the switch hardware the server is connected to, or perhaps even the server hardware itself? Could it be that the server is in the cloud and it's an intermittent Internet connection issue that the ISP is already aware of? Could it be malware that's crept across the company network and is using PCs to conduct as massive Distributed Denial of Service (DDOS) attack on a major corporation, and that as Jeff is using an iPad it hasn't been infected?

It's here that we begin to grasp the complexity of modern PC systems and how interconnected everything is with everything else. Our computers are "always-on" devices, which means they are almost never without a connection to the Internet. This means that a third-party app update service running as a process on your PC is connected to your Wi-Fi network, then to your routers and switch hardware, on to cabling that runs through the building and across town to the telephone exchange, fiber, and satellite connections that connect the exchange to your ISP, more fiber, and satellite connections that link your ISP to the national infrastructure and on to other countries, in one of which sits a server farm containing a copy of your files and documents. That server runs in a virtual machine (VM), running as one of a dozen VMs on one of a thousand rack servers. All of these things are connected together, because they're all making a connection to each other, and communicating pretty much all the time.

As much as this interconnectedness makes computer systems complex on a macro-level, on a micro-level the complexity doesn't go away. We have multiple apps and drivers, which will likely never have been tested long term with one another due to the almost limitless combinations of software and hardware that exist in the world, running on circuit boards and silicon that can contain components as small as 5nm (nanometers) across.

The upshot of all this interconnectedness is that every computer on the planet is theoretically, simultaneously connected to every other computer on the planet. This is how we can instantly access servers in the arctic circle, while simultaneously getting remote access of a PC running a completely different operating system on another continent, download email and receive instant messages from people on the far side of the world, and make a video call to someone elsewhere in our own building.

This is a conceit really, as firewalls, security policies, and structural limitations don't actually mean your Dell laptop is connected to the PC of the finance minister in China. I use it though to highlight how complex our networks can be, and how many devices – PCs, laptops, tablets, smartphones, printers, NAS (Network Attached Storage) drives, servers, datacenters, fitness trackers, PDAs (Personal Digital Assistant), games consoles, televisions, smart speakers, fridge freezers, cars, robots, security cameras, door locks, and Terminators – we're theoretically making a connection to at any one time and of all the things that could possibly be contributing to the problem you face; okay so I'm exaggerating about being connected to Terminators ... or am I?

Summary

We'll talk more about how IT systems are structured, especially in business, in [Chapter 7](#), but this might give you a taster of how complicated the computer systems are that you will face during your career.

In the next chapter though we'll move on to the most scary, unpredictable, and unwelcome part of modern computer systems and look at how they cause problems, rarely provide solutions, and never make your job easy. This being the menace that is the human being.

CHAPTER 3

Understanding Your Users: How Much Do They Know?

It's commonly said that a computer that's left in the box and never used will never have anything go wrong with it. While this is a conceit, it's said to emphasize the point that computers only develop problems because Human beings are using them. It is extremely difficult for a fault to develop with a computer without a Human being involved at some stage.

It's worth noting, as an aside, that the term “bug” was coined by Dr. Grace Murray Hopper, an Admiral in the US Navy when, back in 1947, a moth was found inside a Harvard Mark II computer she and her team had been using and that had developed problems.

So while it's not *always* Humans who cause problems with computers, it's very unlikely you'll find a moth living inside your own computer any time soon. Humans do have an uncanny habit of screwing around with computers, making changes they shouldn't make, and generally believing that they know best. They are, quite frankly, a pain in the butt.

The problem comes from the fact that people have always seen computers in the same way they see any other electronic devices, like their television or microwave oven. While it's true that some (but not all) modern computers *technically* now fall into this category, the ones in the embedded device category such as Android, iOS, and ChromeOS devices, it's still the case that computers of every type are much more complex than the microwave.

The reason for this, of course, is that our computers do much, much more than just warm the baby milk, or cook your TV dinner. Computers are our eyes and ears on the world around us; they entertain, inform, all of us to create and collaborate; and they connect each and every one of us with anybody else, whenever we need to.

The trick then is to tame the humans, which is always much more difficult than it sounds. You can run as many training courses as you like, but you'll always be fighting against three different factors:

- The level of technical comprehension for some people will mean they'll never fully, or completely, understand everything you're telling them. This causes people to switch off and not pay attention.
- Some people will simply be disinterested, think they should be doing something much more important, and they'll switch off too.
- There will always be a few people who will want to tinker with their technology anyway, just because they fancy doing it, or because they believe that they're right.

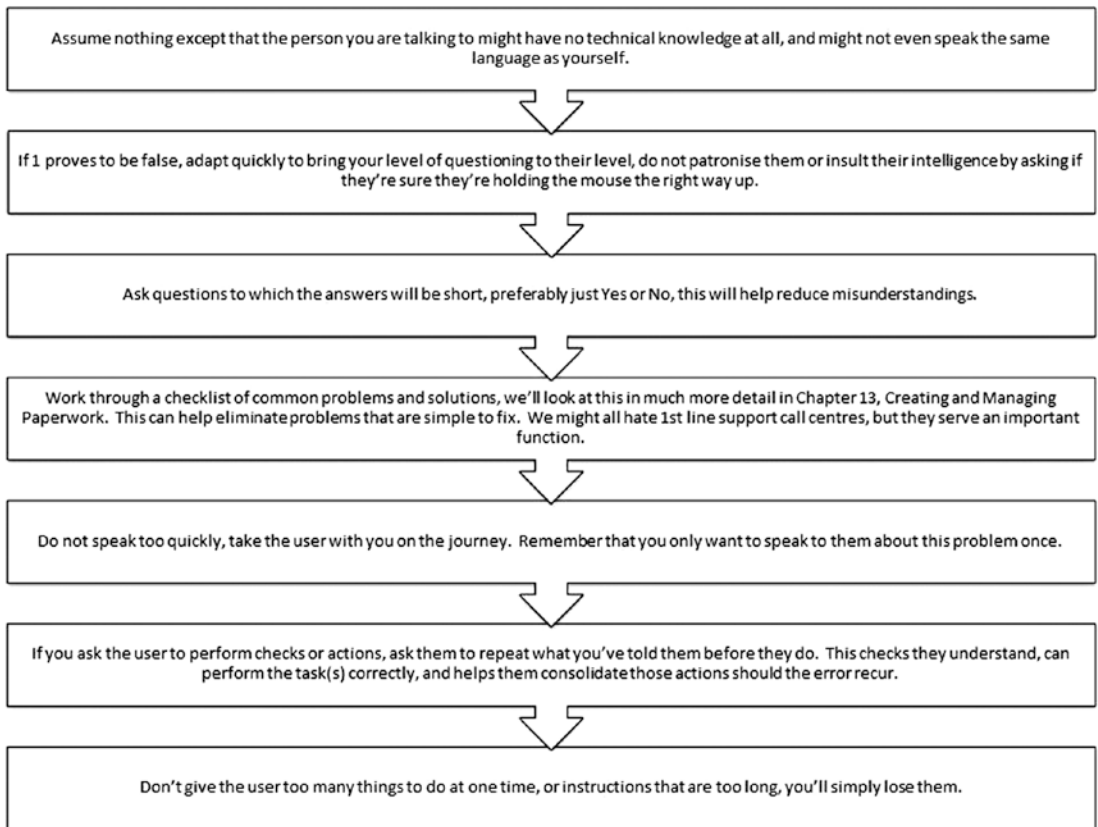
Later in this chapter, I'll look at how you can structure staff training so as to mitigate some of these problems, but first let's look at how you can properly communicate with Humans.

How to Communicate with Humans

Earlier in this book, I covered some of the difficulties you can face dealing with people. Computers are much easier to work with; they all speak the same language, or can at least be configured easily to speak ours, they're reliable and always operate in the same way, and they're always consistent in the messages they give us.

Sadly, Human beings are infallible and infinitely complex. What makes complete sense to one person will make absolutely no sense to many others, and one person being able to accurately and eloquently describe a problem might be completely different to another person being unable to find the words, or the correct terminology when you need them too.

All of this needs to be taken into account when speaking with someone about an IT problem, and I approach things using the following steps:



Following these steps can help avoid confusion, diagnose problems more quickly, and effect solutions more, well ... effectively. Fortunately there are also things you can do with Human beings that can really help and assist you in the diagnostic process, no ... really! In Chapter 16, I will detail tools you can use to enable a user to demonstrate what they were doing at the time the problem occurred, and in chapter 14, *Harnessing the System and Error Reporting in Windows*, I'll show you how to use the Event Reporting system to alert the user whenever an error or problem occurs they may not be immediately aware of.

Most of the information you need to diagnose problems on computers though can be obtained from the computer itself, and this is something I'll detail in Chapters 14 and 15

Sadly it's true that other operating systems, such as Google Chrome and Android, and iOS don't provide this information in any way near the same level of detail as can be found in Microsoft Windows, though where this is available will be mentioned and discussed.

Managing Staff Training

If you run the IT systems in a company of any size, you'll periodically need to train staff in how to use them and in best practice generally. This can involve everything from how to use a particular software package, to how to avoid installing malware on the company network, or being tricked into handing over their passwords.

Anybody who's studied teaching and education will know the basic principles that you need to diagnose, assess, and evaluate the level of understanding that learners have at each stage of the educational process. It's no different when creating staff training, and there are some rules you should follow to create a training course that will be effective, that people will remember, and where they will come away feeling as though they have progressed in their own understanding of the subject.

Learning Theory

There are many different theories when it comes to teaching and education, some of which date back more than a century, some of which are based largely on psychology, and others of which are still taught but are widely ignored by the major universities.

When it comes to teaching in IT support, the ultimate goal is to prevent the problem from recurring. Because it will always be somebody else's computer or technology you are supporting, and because nothing ever goes wrong without a human being involved in some way, it is reasonable to assume that one of the best ways to avoid the recurrence of a problem is to educate the user in such a way as they can either repair the problem themselves if it recurs, or better still avoid it altogether.

For this we use a process of assessment, education, and evaluation. In the first phase, assessment, we are determining the current level of knowledge of the individual. This can be achieved either by asking them to perform a written or online evaluation (though the latter can often provide false results if they also have poor IT skills) or by asking questions, and the latter is far more likely to be the case in IT support.

Essentially you are trying to determine how IT literate the person, or people, is. Do they have knowledge only of what they need, such as using specific functions in Excel and launching and closing apps. If they run a Windows Update manually on their machine, do they actually understand what it is they are doing and why, or is it something that has been advised to them in the past that has become muscle memory?

Assessment is crucial to the process for education, as without it you cannot determine what to teach and how. Does the person prefer to learn by reading, listening, or actually getting their hands on something and doing it practically? You also need to assess their level of understanding when it comes to terminology, as they won't tell you if they don't understand something; they'll simply nod politely and say "Uh huh."

Evaluation at the other end of the process performs a function called *checking learning*, or testing if learning has taken place and the student has understood what you have taught them. Again this can be achieved in a variety of ways, but questioning the learner and asking a selection of pointed and varied questions usually does the trick.

In the middle comes the actual education, and there are some techniques you can use to help the learner understand what you're saying and to properly digest it. The first of these is to ask the learner to repeat what you tell them. This helps you understand if they're getting the terminology correct, and if they've been properly listening to you, or if they've become confused and their mind has drifted.

You can also ask them to repeat what you have shown them. Most people prefer to learn by actually performing tasks; this type of learning is called kinesthetic, and asking a user to perform a task on their computer after you've shown it to them not only helps consolidate that knowledge in their mind but can also provide the person with a small endorphin hit as they feel they have achieved something for themselves.

The next thing you can do as regards teaching is to ask the learner to write down or make notes on what you have taught them. This performs two vital functions. Firstly it provides notes they can later refer back to to help keep the information fresh in their mind (you should always check to see if what they're writing will make sense to them later), but the process of writing down what you're told helps consolidate that information in your mind, as it forces you to actively think about what it is you are putting on paper.

One of the most common learning theories is Maslow's Hierarchy of Needs (see Figure 3-1) that dates from 1943 and describes the theory of Human motivation. If you already teach, you'll probably roll your eyes and groan at this point, as it gets repeated *a lot* and teachers generally get quite bored of Maslow after a while.

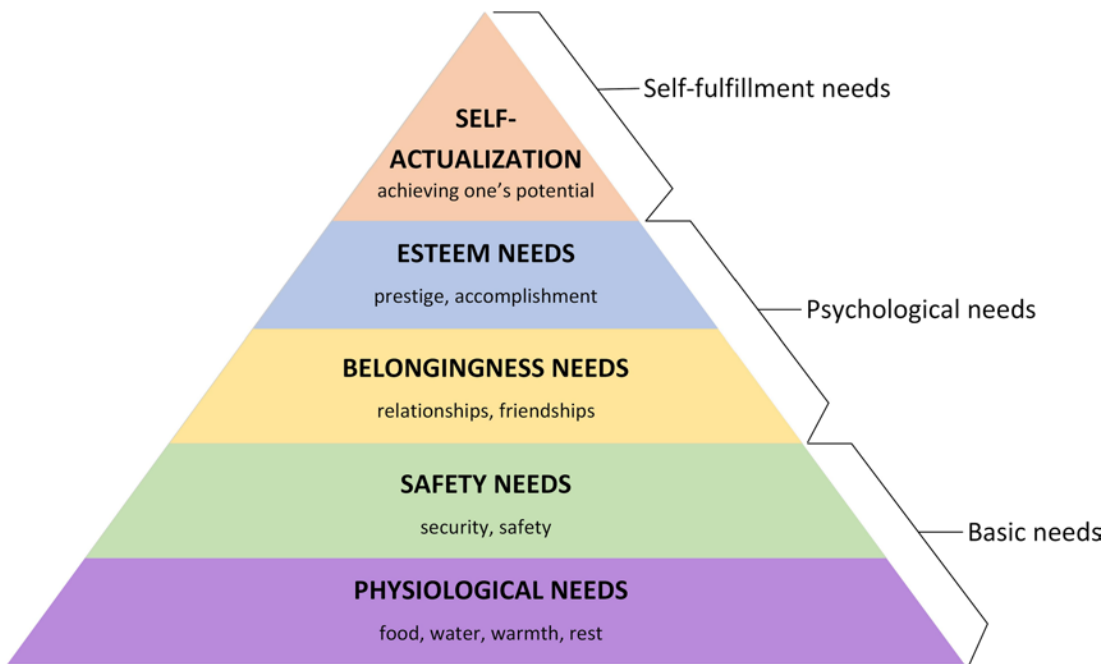


Figure 3-1. *Maslow's Hierarchy of Needs*

This is a learning theory that's based on psychology, but has a good basis in IT support. At the bottom of the triangle are the basic needs, such as food, water, and warmth. These are the first needs the person needs before they can progress themselves in their education and self-actualization.

Once these needs have been fulfilled comes the desire for safety and security, without which we cannot relax and learn. When we are relaxed, the next level of needs is belongingness and friendship. We're now getting into the stage people have achieved when in stable employment, and this is the first stage at which you will encounter people for whom you are providing support.

This leaves us with esteem needs and self-actualization, and it's the esteem needs you are fulfilling when you talk someone through a problem on their computer. The sense of "I've learned something" and the consequential endorphin hit they receive can be a huge motivator for people, and it's the same feeling you receive when you're praised by a colleague, told you're loved by a parent, or receive a like on Facebook.

Self-actualization is achieved when you have fulfilled your potential in a subject or subject area and you can essentially begin to teach it yourself. It's the level people reach when they pass an exam, climb to the very top of the mountain, or set up a new PC for a family member without any help.

Place Everything in Context

One of the most essential parts of teaching and learning is context. If you are being taught something completely out of a context to which you can relate, you will struggle to understand the subject. It is crucial therefore to frame education in the correct context for the learner.

It may already be fairly obvious how you can contextually frame a subject. If the person is having a problem with software they use in their daily work life, framing a context around the tasks they perform, the outcomes of those tasks, and how they then affect other people and other processes can be useful.

As an example of this let's say that a user has a problem on their PC because they thought they'd look for, and manually install a driver for a piece of hardware, and the upshot has been that the hardware no longer works properly. This is an opportunity to explain the interconnectedness of hardware and software on their PC, and how all this then interacts with other systems and servers to which the PC is connected. Everything is tested by professionals to ensure compatibility, and they're always looking at the latest updates, and testing them as well, to make sure they roll out updates at the earliest opportunity.

In another example, somebody might have a network connection problem on their BYOD computer that's preventing them from accessing critical files and documents they need for work. It transpires the reason for this is that they have let their children play with the computer, who have inadvertently installed malware. The security policy on the server has detected this and has automatically blocked the connection to prevent the spread of the malware until it is fixed. In this context there is a clear delimiter between work and home life, professionalism and play.

Having the learner being able to understand how a subject relates to their own life, be that work or home, can help frame the subject so they comprehend it better. Telling someone that the Event Viewer is being configured to display an alert when an error recurs tells the person absolutely nothing useful. Informing them however that when they see this alert, it's letting them know that the problem has recurred and they should down tools and call IT support so they can describe and demonstrate exactly what's happening on their computer at that moment can help them to understand the significance of the alert notification, and they'll be able to see that they're no longer a victim of the problem, they have instead become an integral part of the diagnosis and troubleshooting process. All of this helps boost people's self-esteem, it helps them learn, it encourages them to learn and discover more, and ultimately it helps reduce the likelihood that person will call the IT support helpline with other problems in the future.

Structuring Training and Education

When you design a training course, whether it be just for 1 hour or over several days, there are procedures you should follow in order to make sure everybody can learn, and that you are able to determine if learning has taken place.

Define Your Objectives

What is it that you want to teach? The first step in designing any course effectively is to define clear objectives. What are the things you want people to have learned, or be able to do, by the end of the course? Bullet points here are useful as they can help you separate the course into modules, or even decide over how many hours or days the training will need to last.

If you're teaching people about a new piece of software, what do they need to be able to do by the end of the training? They might need to be able to sign into the software, this will be step one, but not necessarily where you start (I'll come to this in a bit). They might need to understand the menus and options available to them, how to process data through the software, and how to check that data is correct before sending it to another department for their consideration.

Each of these is a separate objective and should be treated as its own self-contained module. The reason for this is that when you speak to anybody, be it in a training room or at a conference, there is a natural Human tendency to begin to drift off after 5 minutes. After 15 minutes you will have lost some people completely.

Mix Things Up a Little

Because people's minds can drift, quite naturally, after a few minutes, there are different things you can do to keep people's attention. You can prod people and wake them up occasionally by using techniques such as asking them questions, changing the method of your training (such as inserting a short video into the course), or asking them to perform an activity.

Performing activities is a good way to enable people to consolidate what you have taught them, and for you to be able to evaluate what and how much they understand. I'll come to these subjects in more detail shortly, but it's always good to think about what activities, or additional components (such as videos), you can insert into your course.

Assess Your Learners' Knowledge

Before you even begin teaching an individual, or a group of people, it is essential to assess what knowledge they already have and what they already understand. You might find that there are people in your training session who already have a very good understanding of the subject matter, and there could be others with no experience of the subject or technical knowledge at all.

You can assess learners in different ways. You can ask them questions, though this is best done via a short questionnaire, and not by direct questioning. The reason for this is that if you have even just a single learner who has no understanding of the subject, they could feel as though they are “stupid” compared to the other people in the group. If this happens, you will have lost that learner immediately, as they’ll be far more preoccupied with their own embarrassment than with whatever you’re teaching them.

It is up to you to assess the learner’s prior knowledge, don’t allow that to single out anybody, and make sure that you get feedback from everybody in the room. You can then use this assessment to adapt your teaching *on the fly* so as to provide the best education possible.

Use Mixed Peer Groups

When you teach, you will almost always have people of mixed abilities in the class. There will be, as I have already mentioned, some people who already have a good understanding of the subject and others who have very little knowledge.

You can use this in your training when asking questions of the learners. I’ll talk more about consolidating learning soon, but if you want to ask a group of people questions about what you have just taught them, you can pitch the level of the question differently for different learners.

To them, the questions will appear completely random, but what you’re really doing is asking more technical questions of the more knowledgeable learners and more entry-level questions at the beginners.

When you give people activities, you can use mixed peer groups to help people learn. A mixed peer group is one that contains people of different abilities and with different types of skills. If you place four people in a group where one of them is already knowledgeable, but another isn’t, they will almost always pass knowledge between them, and somebody who is learning, or who has just learned a subject, can often phrase or describe even complex concepts in easy to understand ways that you may not have considered.

You can also create groups where people have different skill sets, as this can help learning take place for everybody in the group. Having somebody who has good organizational skills, or one with experience of the software or hardware the company has been using before, can often help everybody contextualize what it is you are teaching them.

Help the Learners Consolidate What They Have Learned

Speaking of contextualizing the subject, you need to give the learners opportunities to consolidate what they have learned. It's good to do this on a regular basis, as this means they can take a chunk of a subject, and get their head around that, before moving on to the next chunk of the subject.

Practical or group activities are a good way to help people consolidate what they have learned. If you're teaching software subjects, for example, and have access to computers with that software in the room, then asking the learners to work through the process you've just taught can very often help them contextualize it.

This feeds into how most people learn, and this is something called learning styles. There are four different learning styles that people have:

- **Visual** - Where people learn best by reading or looking at pictures and diagrams
- **Auditory** - Where people learn by listening to things being explained to them
- **Read and write** - Where people learn from reading books or notes, and making their own notes based on what they have read
- **Kinesthetic** - Where people learn by performing tasks or actions

Kinesthetic learning is by far the most common learning style, so giving learners tasks where they get involved with the subject and actually perform actions on hardware or software, or non-computer group projects based around the subject, they can very quickly consolidate and better understand what they have just learned. This feed back into what I said earlier in this chapter about always contextualizing the subject for the learners.

Evaluate the Learners

When you have finished each module of the learning, and again at the end of the course, you need to evaluate the learners to determine what they have learned and even if learning has taken place.

A common way to do this is through a questionnaire, or through individual questions (again each one being specifically for the level of each individual learner). You could also set them a task or project to complete, such as perform one or two tasks in a piece of software, or change a component in a computer or other pieces of hardware.

Whatever you do, evaluation is an essential part of the process. Only by properly evaluating the learners can you determine if learning has taken place, and if further or additional training might be required.

Écouter et Répéter

Speaking of further training, you might find it a good idea to bring the learner or the group together again after a week or two, to evaluate them again, and to plug any remaining gaps in their knowledge.

You might remember if you learned foreign languages in school that you're asked to listen to, and repeat what you're told by the teacher. It's no different when bringing learners back together for a refresher.

You would normally begin with a quick overview and reminder of what it is you have taught them, followed by asking them some questions so that you can see how much they still remember.

This method allows you to quickly identify any skills gaps, so that you can then provide additional tutelage to fill those gaps, and allow each learner to consolidate their learning further.

Summary

No two people are the same. Everybody is different and individual in some way. This could be their level of technical understanding or experience, it could be their desire to meddle or leave well enough alone, it could be their primary language and location in the world, or it could be one of any number of different things.

It is very important therefore to never make assumptions about the people you support; what they know, what they understand, how they comprehend different things, and so on. Treating everybody as a unique individual, and helping each one to learn at their own level and at their own pace, will make you a great teacher, a great support, and will ultimately help reduce the number of calls you receive in support.

Helping people consolidate things in their own mind is one thing though, but wrapping your own head around complex troubleshooting problems can be a challenge in itself. Over the next three chapters, we'll look at this subject, and you can quickly understand what is causing the problem you face.

PART II

IT Support Methodology

CHAPTER 4

Flow Logic and Troubleshooting

People often believe that IT support is about sitting down in front of a PC, printer, or other device, finding the problem, making some scornful comment about how easily this could have been avoided, and then fixing it. In truth this is a very outdated view and doesn't relate to the support industry as it exists today.

I have two careers, and both of them feed into one another and work in tandem. I'm the author of tech books and courseware, with a particular focus on troubleshooting, but I'm also a teacher of English and Maths. For years I've taught teenagers who have left school with no qualifications and the long-term unemployed to help give them the skills they need to break free from poverty; to get into, or back into the job market; and to prosper on their own. It's hugely rewarding.

I knew back when I was in school that I would like to teach but rather fell into a career in IT instead, just because my proficiency reduced the barriers to entry for me. It wasn't until my mid-30s I began to teach professionally and I learned a lot.

How Does Flow Logic Work in Troubleshooting?

In Chapter 1 I talked about the process of troubleshooting and the three questions I use as the basis of any diagnosis, What, When, and How. I also talked about using checklists to quickly eliminate problem areas and to help the person providing support to understand the language, terminology, and level of understanding of the user.

We'll look at these checklists in detail in Chapter 13, *Creating and Managing Paperwork*, but it's important first to look at the process of troubleshooting as it's far more complicated than just find the error ► fix the error.

It's logical to start with the question "What is the problem?" You need to understand the basics of what it is you are facing before you can progress to finding a cause and ultimately a solution.

Then follow two more critical questions ... "Has the problem occurred before?" and "Is the problem affecting anybody else?" Both of these give you essential information. The former because it may be a support ticket has been raised for this issue in the past, either for this user or another user, and the solution to the problem is already documented. It could also be that the user knows they have caused the problem, what it is, but that they're a little embarrassed and therefore not forthcoming with all the information you need, including how it was fixed the last time.

The latter because if you are troubleshooting in a work environment, it immediately tells you if the problem is localized to the one computer or device, or if it resides elsewhere on another device or devices, or on the network.

Process of Elimination

Discovering what is causing a problem is helped considerably if you can eliminate what definitely is **not** causing the problem. This might seem a little odd to be working at the process backward as there are far more things that won't be causing a problem than the one thing that is, but there is logic to taking this approach.

Let's take the example of a faulty network connection so as to demonstrate how a process of elimination works well in troubleshooting. A laptop is unable to make a connection to the Internet. This could be a result of a dozen different factors, so we ask questions and test in a methodical way.

1. Are there any other devices nearby, and can they access the network and the Internet?

If there are any other devices around, and this can include anything from a desktop PC to a security camera, this question tells us if the problem is local to the device on which the problem has been reported, or if it may sit elsewhere.

2. Can the device see the local network?

This question again helps to tell us if the problem is local to the device or elsewhere. For example, being able to see the local network but not the Internet could mean there's an outage with the ISP.

3. How is the device connecting to the network and the Internet?

This is where we begin physical tests that are all quick and straightforward to do. If the device is connecting via Wi-Fi, what happens if you connect it directly to the router using an Ethernet cable? For this purpose it is always a good idea to carry a length of Ethernet cable and a USB to Ethernet adapter.

This test can help narrow down the process further. If the device connects to the network and the Internet via Ethernet, then it could be one of several, but only a few, issues ...

- a. The Wi-Fi driver needs installing or updating.
- b. The settings for the Wi-Fi network are incorrect or have become corrupt.
- c. The Wi-Fi settings in the router have changed, or Wi-Fi is unavailable in the router for some reason.

All of these three options are simple and straightforward to test, but because we have approached this in a methodical, procedural way, we are able to get to a satisfactory resolution far more quickly than if we made assumptions from the outset and just reinstalled the Wi-Fi driver.

What we've done is determine, through a process of elimination, what is definitely **not** causing the problem. It is **not** the Internet connection, it is **not** the networking stack on the device, and it is **not** the device hardware itself. With these eliminated we can focus our minds on the few number of things it actually is, or still might be.

“Information Is All”

In the James Bond movie *Spectre* (Eon Productions, 2015), the villain Ernst Stavro Blofeld secretly builds an intelligence network that he convinces nine of the world's largest economies to join, including the United Kingdom, the United States, Canada, Mexico, and China. This “Nine Eyes” alliance would have given S.P.E.C.T.R.E. (The Special Executive for Counterintelligence, Terrorism, Revenge and Extortion) unprecedented access to the security and intelligence information of all member states, making their police and security services easy to counteract.

The Nine Eyes alliance is actually based on a real security and intelligence project. Five Eyes is an intelligence sharing agreement between the United Kingdom, the United States, Canada, Australia, and New Zealand. The foundation of Five Eyes can be traced back to the end of the Second World War, and it has proven to be a controversial communications monitoring project that has allowed its members to spy on one another's citizens.

That's an interesting aside, but it demonstrates that even James Bond villains are prepared to hatch an evil scheme based entirely upon collecting and sifting through huge volumes of data. "Information is all" as Blofeld correctly states in the movie and having access to the correct information when you provide IT support (I would imagine *Spectre* would have employed its own evil IT support personnel) can simplify the task of identifying and resolving a problem considerably.

The information you have access to will vary considerably depending on the type of organization you work for and who it is you provide support to. If you are self-employed and provide support for people in their homes, or in your own shop, then you probably won't already keep records of client work, though you may decide to change this policy after reading Chapter 13, *Creating and Managing Paperwork*, but if you work for a large IT provider, as I did with Fujitsu Siemens, the amount of available data will be pretty large.

This information will include the asset tag and serial number for each individual piece of hardware, from keyboards to servers, records of every call received, by whom, what about, and how it was resolved. Who it is in your organization that dealt with issues brought to you by clients, and more besides.

Shortly after I started working at Fujitsu Siemens, it was this information that quickly led to the diagnosis of a major rollout of Dell computers that had gone horribly wrong. The GX270 desktop, which was a pretty average, everyday workhorse had been built with capacitors that were of a quality far below Dell's usual high standards and consequently were exploding and spilling chemicals onto the motherboard and other components inside the case which then rendered the PC useless.

When this began, we were getting around 10 to 20 calls a day from bank branches around the United Kingdom, and it was because of the record keeping that we were quickly able to identify what the problem was, what model of PC it affected, and where those PCs had been installed. This led to the PCs being temporarily decommissioned, while engineers could replace the motherboards to prevent damage to other components inside the PC's case.

The upshot of all this was that any call our department received about a GX270 could be dealt with in less than 2 minutes by a simple search on the asset tag and date of installation data. This hugely reduced the workload on an already busy IT support department (but I would say that wouldn't I) and helped prevent further calls from being received. Ultimately this best practice procedure saved both Dell and the bank hundreds of thousands of pounds and helped cement our own reputation.

Once you begin collecting the correct type of information on the support you provide and set up processes to collate and manage that data in the right way, you will find it simplifies the role of the person providing support considerably.

We will look how we collate the correct information that we need in Chapter 11, *Why Good Documentation Matters*, but the root of collecting this information is in the end user. These people are your eyes and ears; they are your best and initially your *only* source of information and data, and one you need to harness effectively.

Paperwork Is a Pain, or Is It?

I hate paperwork as much as the next person; actually being a teacher I probably hate it more given how often information has to be repeated ad nauseum in order to fulfill the requirements of a few struffy-haired managers with nothing better to do than pile unnecessary administration onto already overworked staff ... but I digress.

It's a good point though the paperwork is a pain in the butt. I can't think of anybody who enjoys it but largely it's there for a very important reason, and when it comes to IT support, it's incredibly useful and important.

I'll detail how to create and structure paperwork thoroughly in Chapters 11 to 13 (I know you can't wait). Paperwork in IT support can help support flow logic, it can help ensure that everything is covered and nothing is missed, and it can make sure that repairs and fixes take place quickly, efficiently, and cheaply, no matter who does it or when.

Begin at the End, but Don't Work Your Way Backward

You might think that it's always important to start a process at the beginning, but with IT support you're always forced to start at the end. There is a problem, and you need to fix it. It's your job to find out what it is that brought you to this outcome; it's the detective work I talked about.

You might be surprised then to hear that you don't work your way backward to solve problems. Yes, you want to find out what caused the problem, but this might not come at the end of the diagnostic process, it might come near the end, or even half way through.

You'll find yourself jumping around when diagnosing problems, and this is why following a logical diagnostic and recording process is so crucial. This means that when you find the cause of the problem, you shouldn't necessarily stop examining what happened.

Knowing what the problem is, and how to fix it, is great, but what's better is understanding *why* and *how* that problem occurred in the first place. How do you know the problem won't recur again, perhaps even tomorrow or as soon as you walk out of the door, or that it won't happen to somebody else in the building?

“Don't Stop Thinking About Tomorrow”

The solution to a problem is only ever half of the story. The cause of the problem is the rest. If you know the cause, you are so much better equipped to provide long-term support, or perhaps find that you don't need to provide support for the same or similar problems again.

It's all too easy to say “Aah, it's this that's causing the problem, so I'll just fix it in that way and we're done.” Who does this help? It's like putting a cast on somebody's leg without first realizing that they can't walk because the bone is broken.

It is just as essential to understand the cause of a problem as it is to understand how to fix the problem. You might find that the cause is something that potentially affects dozens, perhaps thousands of people, and that this and similar problems can be averted by changing policies or procedures slightly, or by tweaking settings in software.

This is why you should never stop at the problem; you wouldn't be the best support person you can be if you did. The more you can understand, the more you can learn, the more you can help, and the more people you can help.

The Impossible Is Possible

If there's one last aspect to flow logic when troubleshooting problems, it's actually the most important one of them all. Always, really always avoid skipping over things in order to determine the cause of a problem. This doesn't mean that you'd reset a smartphone in the way first-line support will tell you to do, when they're helping you diagnose a signal problem; that helps absolutely nobody most of the time, but it does mean that you shouldn't dismiss anything as being impossible.

Nothing is impossible in this world, there might be things that are very, very improbable, but impossible they're not. This is hugely important to the flow of diagnosing problems, and dismissing something as "definitely not being the cause" can not only result in your overlooking something that later turns out to be important, but if you need to hand the case over to another person, such as an engineer, they won't have all the information they need to fix the problem, quickly and efficiently.

This rather brings me back to the paperwork; it always seems to come back to paperwork. Checklists are your friend. They'll help make sure that you miss nothing and that you overlook nothing. They'll help ensure that you don't put anybody in a position where they'll misunderstand the problem, or ask the end user to do something unnecessary, or that they've already done or been asked to do.

Work with the Team

Armed with well-structured paperwork, and a well-structured approach and methodology, you will not just provide quality IT support, but you'll also work well within a team of people who provide support. You might be hit by a bus tomorrow after all (though I hope you don't) or be off work sick for several weeks.

Whether you're unwell, on annual leave, or just someone who has to pass jobs on to field engineers, you are part of a team. This makes you part of the process, a cog in the machine and not the machine itself. There might be a great deal of satisfaction in finding the solution to a problem, and there might even be a temptation to keep some things to yourself on occasion for personal benefit, but you should always remember that the team comes first.

If this seems like preaching to the converted, even though you may not always achieve it, just think back to the last time somebody passed a task on to you without properly documenting it, or without adequately briefing you first, and remember who you felt; we've all been there at some point or another.

You might not even like the people you work with, they might not like you for that matter, but you all work together. The way you respond to and communicate with each other is crucial to the success not just of yourself but of the whole unit, your department, and perhaps even your company.

Summary

Working through a problem always means beginning at the end, and always means ending at the beginning, that is, if you do things properly and have good and logical procedures in place. When you're designing flow processes and procedures for yourself, your team, or your business to work through, bear in mind that at any point, anybody needs to pick up the job you're working on and be able to immediately run with it.

In the next chapter, we'll take this to the next logical step and look at how we can use this logical approach to properly and appropriately ask end users, who may be very non-technical, what the problem(s) are they they're facing, whether you're in the room with them, on the phone to them, or chatting to them online.

CHAPTER 5

Querying Users Effectively

We've already established that most computer users aren't technically minded. This is a good thing because it's why you and I have jobs. It can be all too easy though to make assumptions about people, speak to them in your own language, ask them questions they'll never be able to understand, and just confuse them entirely.

This of course helps nobody, it doesn't help you and it doesn't help them. It becomes especially true if you work in a call center in a country different from the one in which the user you are supporting is based, or from.

How to Query Users Effectively to Diagnose Problems

Call centers in India, for example, are staffed by people who speak English primarily as a second language taught to them in school. You might think then that they would be well placed to provide easy to understand IT support to English-speaking countries such as the United Kingdom, the United States, Canada, New Zealand, Australia, and South Africa. However it's here we hit a language barrier as it's very common for people who learn a second language to use language forms such as verb-subject agreement in the form of their first (primary) language. It's the way that second and additional languages are learned and spoken that can make calling foreign call centers a difficult experience, that can sometimes make foreign press conferences tough to understand, but that also made Yoda so funny and enjoyable to watch in *Star Wars*.

This problem isn't just confined to people who speak second languages, not by a long chalk. Regional dialects can cause problems as well, both because accents can often be quite heavy and also because vernacular can vary from one region of a country to

another. If you don't believe me, just ask somebody from Scotland how they get on with Siri, or their Amazon Echo, if they manage to get it to understand them at all.

Hinglish (a mix of Hindi and English) and Chinglish (a mix of Chinese and English) are often mocked by comedians, but they represent the problems you can face in IT support pretty well. In India, for example, there are 22 major languages spoken with a total of over 720 different dialects. The United Kingdom has 28 separate dialects just for the people speaking English.

As an example of how language changes from one part of a country to another, shortly before I worked for Fujitsu Siemens I had moved from the South East of the United Kingdom to Yorkshire, just above the midlands and a couple of hundred miles from where I'd started. I'd found it difficult to buy a sandwich that was made as filling between two slices of bread, as every sandwich shop seemed to sell them in a bun.

Of course the people of Yorkshire don't all like to call it a bun as there are several different dialects spoken in that region alone, and in one moment of utter hilarity (that made me deeply unpopular with my boss I might add), I stood on my chair in an open-plan office of 280 support personnel and asked the question "Is it a bun, a bap, or a breadcake?" The arguments were still raging 2 hours later with some even suggesting it was definitely a cob.

Users Can Be Anyone, and Anywhere

So you think you know who you're supporting do you? Perhaps you're offering support to a major bank as I was, or maybe you're offering support on behalf of a major technology company, like Dell or Samsung. Alternatively you could be working in a small independent shop, or perhaps in a large chain such as an Apple or Microsoft store in a diverse and multicultural city. Whatever the circumstances in which you find yourself, the problems you face will likely be the same.

The differences between a customer standing in front of you pointing at their dead laptop and a worker from Germany who is on a crucial business trip to Norway are slight, and only truly differentiated through the face-to-face encounter giving you the added benefit of facial expressions, body language, and a handy finger that can be pointed at the bit that doesn't work.

People also migrate from one part of the world to another, or work on secondment in a foreign division of the company. People even, shock horror, sometimes need local tech support for their devices when they're on vacation.

Never Make Assumptions

I always tell people who teach to never make assumptions about the people they are teaching. Unless and until you have performed substantive initial assessments with them, there's simply no way to fully gauge to what extent they understand the language you speak, what their levels of literacy and numeracy are, what their level of technical expertise is, how and even *if* that expertise stretches to the technology you're helping them with, or whether they had an especially heavy night yesterday and didn't get a lot of sleep.

You will also not know how busy they are, or how important this piece of technology actually is to them. It might be that the device is almost brand new, and the user is frustrated because they spent more than \$2,000 of their own money on something that suddenly doesn't work. Conversely it might be the last day somebody has on a business trip and there's a deal to finalize (in which case some lateral thinking and problem solving is always a good approach), or they might have a big presentation to give in 3 hours.

The human factor can also rear its ugly head when providing support. It's entirely likely that the person who contacts you is polite, courteous, and patient at least most of the time. On the other hand, they may be grumpy, impatient, impertinent, frustrated, and perhaps even downright rude.

Yes or No?

This last reason is probably the best one for keeping things brief and asking short, concise questions that each require a short and concise answer. To achieve this well takes practice and you really *do* need to know your subject area. Their interpersonal skills might be awful, but yours need to be brilliant, all of the time. This is as much for your own benefit as it is for theirs or anybody else's.

Not every question can be answered with a simple yes or no answer, but many can be answered more quickly and understood more readily if presented in the right way. There is, for example, a big difference between the questions "What is the asset tag for your PC?" and "There is an asset tag number printed on a red and silver sticker somewhere on the main case for the PC, can you read it to me please?"

Even more technical questions can be presented in straightforward and easy to understand ways. You can improve upon “can you check the monitor cables?” by rephrasing it as “There are two cables in the back of the monitor, a power cable that’s connected to the mains electricity, and a video cable that is connected to the main PC box, can you check they’re all seated and plugged in properly please, but please be safe when you’re doing it?”

When you need to get really technical, this can be achieved with relative simplicity too. “Can you open a Command Prompt window and type ping 192.168.1.1?” is much less understandable than “Can you hold down the Windows key and then press X on your keyboard, then in the menu that appears click ‘Command Prompt’ and when that a new window appears type PING ‘P.I.N.G.’ followed by a space, then the numbers 192 then a dot, no space 168 then a dot again, 1 dot 1 and press the Enter key?”

While instructions require longer and more complex statements and questions, you might be surprised what you can ask to get a simple yes or no answer. “Can you describe the network problem to me?” could very well result in confusion, incorrect terminology being used, and the wrong message being delivered. Instead you can break this down.

“Are you able to access web sites, or is it just files on the network you can’t access” will elicit a simple yes or no response, as will “Have you tried accessing the same files from your laptop, or from another PC?” Always try and break the processes you want the user to work through into smaller, more manageable steps. You understand what you want them to do, but if they knew how to do it, they would have probably done it already. Smaller and more manageable steps are also much more likely to elicit a simple yes or no answer, bypassing all the problems that can arise from people calling the ethernet cable a USB wire.

Take the User with You on the Journey

If you’ve ever heard the term “First World Problem,” then you’ll no doubt be able to relate to what I’m about to say; that it can be traumatic to be separated from your computer and the Internet.

Genuinely, it’s such a big problem that you have to wonder what people did, how they communicated, and how they worked, shopped, and enjoyed themselves before the iPad was invented.

I know this sounds like I'm being glib, but it can cause genuine anxiety when people can't get online, so part of your role as an IT Support person is to reassure them. Why should you do this? Frankly you should do this because it's in your best interests to do so.

Have a think to how you or the people you know react when they suddenly find they can't get online, or they can't do the work that needs to be done. You'll probably find that at least some of the time they worry. Worrying directly affects the concentration somebody can give to whatever they have going on that day. This includes how helpful and patient they can be for what you're doing to help them, and how much information and the quality of that information they can provide.

It's no fun trying to repair somebody's computer if they're either too distracted to give you clear and precise answers to the questions you need to ask, or if they're constantly badgering you about how long things will take.

What can prove very useful then is to take the user along with you on the journey. This doesn't mean bundling them into the passenger seat when you take their laptop away for repair, there could be safeguarding issues involved with that, what I mean is that if you keep the user informed of what's happening, why it's happening, and how long the process is likely to take, they will be more patient, more understanding, and much more aware of how they might be able to prevent the same or similar problems recurring in the future.

The User Is Your Friend ... Yes, Really

Taking the user with you on the journey rather implies that you have to be nice to them. Lots of people work in IT because they're not *really* people, people. I would count myself among this number. The amount of introverts working in IT is very high, and disproportionate to a lot of other industries.

Sadly there are many times when you actually do need to interact with other people in the real world, I know, it's depressing. You do then really need, if not be a people person yourself, to be aware of who the other person is. Let me explain.

There are, as I said at the beginning of this chapter, a huge amount of different types of people and personalities in the world. These include non-technical people, people with poor communication skills, people who have had a poor education, people who have had a private education and who may consider themselves superior to others, people who lead, people who follow, extroverts, introverts, people with long attention spans, people with no attention span, and so on.

You need then to be able to speak to people if not on their own level, on a level that they can understand and feel comfortable with. There's no point talking technical with someone who doesn't understand what you're saying, or discussing with somebody the importance of maintaining security vigilance across the department when they only consider themselves to be a small cog in the machine.

I mentioned previously that you should never make assumptions about people, about who they are, what they understand, how they think about problems and approach the world. Being able to get a handle on who a person is, even roughly, when you begin working with them though can be a great way to help things along and make your own life and task easier.

Swipe Left or Swipe Right?

As much as we might want it to be the case, real life is not the same as online dating. If you don't like somebody, you can't just swipe left to get rid of them, you're stuck with them.

Does this mean you have to like everybody that you meet and work with? Of course it doesn't. What this mean though is that you have to be as nice to them, as you would be to someone you would definitely like to swipe right on.

Why do you need to treat everybody as though they're the most important person in your life? Well let me just say this. If this is a problem you're facing that's preventable, or that the user can fix themselves in the future, and they're an annoying or otherwise unlikable person, then being super-nice to them might mean that you never have to see them again!

I'm not kidding about this either. The whole point of providing IT support isn't to fix problems with technology (though obviously that's a big part of it), it's to reduce your future workload and to make things easier on yourself. There's the obvious downside to this, that in doing so you'll get a great reputation which will create recommendations and drive more customers to you, but we don't live in a perfect world.

Politeness costs nothing, and in the world of IT support, it really is its own reward. Snarling and griping at a customer just because they've gone and done something really stupid ... again, is not going to prevent them from continuing to be stupid in the future. Explaining to them, on their own level and on their own terms what happened, what they did wrong, and why they should never do it again is a much better approach overall.

The Non-technical Dictionary

If you searched Wikipedia for the word iPad, then you'd get the result "iPad is a line of tablet computers designed, developed and marketed by Apple Inc.," and if you searched for the term desktop computer, you would see in its description, "The most common configuration has a case that houses the power supply, motherboard, disk storage; a keyboard and mouse for input."

If you ask the average man or woman in a workplace what an iPad was, however, they'd very probably point to their Samsung tablet running Android, or their Surface Pro. If you asked them to point to their desktop computer, they'd almost always point at the monitor. Learning the non-technical dictionary then is a hugely useful skill, because it means that you can avoid confusion in the first instance by changing your language accordingly.

How can you do this? Asking a user to turn their tablet over and tell you what the manufacturer's logo on the back looks like is one way, asking the user where the big box that has the power button for their PC is another.

You will be able to query users much more effectively, and get a much higher quality of information, quickly and easily, if you're prepared to use language they understand. If they then immediately come back and say that the tablet is a Samsung Galaxy running Android, then you will know immediately that you can change the language you use appropriately. If on the other hand they can't find the logo on the back, because their tablet is perhaps in a leather case they've not removed it from, then you need to take this into account too.

Online Chat

This brings us on to the subject of online chats and remote support. With better broadband connections, and the advent of always-connected PCs, this is becoming more and more common. You might not be trying to fix a problem over chat, you might instead just be trying to obtain information about the problem first, but the rules for using online chat in IT support are the same, and they all feed back into what I've said so far in this chapter.

Patience and yes no answers here are the key. How good is the end user at typing, and what type of keyboard are they using? If they're using a smartphone, and there's no way for you to know whether they are or not unless you ask them directly, then they certainly won't be typing long answers for you.

If the user has a low battery on their device, or is perhaps trying to save their battery as they have a long day ahead of them, they won't want to be giving you long answers either.

They might be somebody who has a motor problem, or poor eyesight, making typing difficult for them, or there could be other problems.

Reassuring people occasionally can help too, and you'll very often in the online support chats you instigate yourself see people replying with standard, pre-scripted replies such as "Not to worry, we're here to help" or "Don't worry, we'll get that sorted for you." This is simple psychology that servers to reassure the person seeking support that you understand their needs, as well as letting them know that you're still there and haven't forgotten about them.

So when you're chatting online with somebody about a problem, keep things as brief as you can while also getting as much information as you can. You also need to keep things relevant, as if you chat to someone as though they're a complete noob, when it's clear they have more technical knowledge than that, they will very often lie to you about things.

"Yes, I've reset the device" might be a very common response when they haven't reset their device at all and have no intention of doing so because they're certain that won't fix the problem (even if it will). So you need to be prepared to be lied too, which also applies when dealing with people over the phone.

It's conscious and deliberate questioning that will always win the day in this scenario. While people can often and quickly expand on what they might mean when you're taking to them on the phone, over chat, this simply won't be the case. The shorter your questions, the shorter the required answers, and the more quickly you can get the information you need, the better the outcome will be.

Summary

There are all sorts of people, with different likes, experiences, and skill sets. This means you need to be able to quickly judge who a person is, what they might be able to tell you, and how you should best question them. This is a skill in itself that can take some years to properly hone. The most important fundamental though is never to make assumptions about anybody. You know nothing about them when you first start talking to them, and so no assumptions about them should be drawn until you have more information.

Speaking of more information, in the next chapter we'll take everything I've been talking about in these first few chapters and draw it all together to see how we can join the dots and correctly diagnose any type of IT system problem.

CHAPTER 6

Joining the Dots: Finding the Root Cause of an IT Issue

The biggest selling item at most airports around the world for people traveling on vacation is a good paperback thriller. People love detectives and detective work. This is because human beings, fundamentally, are explorers and adventurers who want to learn about new cultures, new experiences, and about the world around them and the universe around that. This is the primary reason why Scandinavian drama has so gripped the world in recent years and why Sir David Attenborough remains even more popular today than he was when his ground-breaking *Life on Earth* (BBC) series was first broadcast in 1979.

This desire to learn and explore is what separates mankind from the animal kingdom. You don't see many cows wanting to venture out of their fields to learn about the pond of frog spawn a mile or so up the road, and elephants and rhino in Africa haven't formed a peace pact so they can gang up on hunters and poachers ... at least not yet, but there's still time with this one.

This is your biggest advantage in IT support, both with yourself and with the people you're helping. For yourself this means that everything becomes a puzzle; you can work through the puzzle to solve it, work on your own or collaboratively with colleagues or the end user, and gain satisfaction (and that endorphin hit I mentioned earlier) when the issue is resolved.

The end user on the other hand can get an endorphin hit of their own when they learn what it is that happened, or what it is they had done wrong, and find out how they can fix it. They can receive yet another endorphin hit when time comes to fix it again, or help somebody else with the same or a similar problem.

This brings us back to Sherlock Holmes, and eliminating the impossible. Think of IT support as a big checklist of possible items, under each of which is a long checklist of possible sub-items, and under each of those it's repeated again and again until all the possible causes are exhausted.

If you can eliminate the items at the very top of the list, you automatically also eliminate every item under those, and every item under those. This may take two, three, or four steps to accomplish, but it works much better than making an assumption, ducking in half way down the list and trying something out. You might be right, it could be a very well-played educated guess, but it's not methodical and can eventually lead to bad and sloppy practice.

To give an example of this, we'll say that somebody can't synchronize their email to their device. If you ask them to access their email in a web browser, you can immediately eliminate all the possibilities associated with the email service being faulty.

If you then try a different email account on the device, or try and access their email on another device, you can eliminate problems with the users' own device, or tie the problem to that device specifically. From there you have only a limited amount of possibilities such as a corrupt or incorrect configuration in the email app, a corrupt email app itself that might (for example) have just received an update, or a networking issue on the device. To get to this point will have just taken you a couple of minutes.

In another example, you've been brought a laptop because nothing happens when the user plugs devices into the USB ports. The first thing to check is to plug something into *every* USB port to see if they've actually tested them all and to see if they've only tested one peripheral in them. If it's a desktop PC or an all-in-one, there might be ports around the back they're not aware of. This will immediately tell you if it's either a hardware or a driver issue, if there really isn't any issue at all, or if it's the USB peripheral which is faulty.

Lastly somebody is complaining that their PC won't start. Step one is to check if the cables or the battery are seated properly; step two is to check for lights, fans, and any tell-tale whirring sounds; and step three is to see what happens when you press and hold the power button for 4 to 10 seconds (depending on the device).

This is a good example, as the sheer number of problems that can prevent a device from booting is staggeringly large, and contains everything from a blown fuse to a corrupt security partition.

Use small steps and small checks to eliminate the possible causes of a problem one by one, each time crossing off large numbers of boxes on your checklist of infinite possibilities until you eventually have only two, three, or four possible causes left. You might be surprised how easy this is, and how quickly you can get there.

The Beginning of the End

When you begin diagnosing an IT problem, you already have information, and it's always with that information that you should begin your diagnostic process. This of course details where you're at, at the end of the process that led to the problem.

This means you will know what the outcome of the events leading to the problem is. Does something not work, will something not connect, and so on.

It might not seem incredibly helpful to begin at the end, but it's just like any murder mystery where the detective only has a body. They need to find the murder weapon, discover the motive, and then the clues leading back to the eventual discovery and unmasking of the killer; it can be quite fun.

So what do you have at the end of the process? You need to break a problem down into its individual components, and deal with them one at a time, as you don't know if there might be interconnected problems causing the issue, which isn't uncommon.

Working Backward

Let's take the example of something that isn't working. It doesn't matter in this example if this something is software or hardware, as the initial questions you'd ask, and the investigation will progress in largely the same way.

The thing isn't working. So you need to know is it the *only* thing that isn't working? Are there other things on this PC or hardware device that also aren't working? Are there any other PCs or devices on which this thing isn't working? Are there reports of the same thing not working outside of the building and on other premises?

These first few questions can get you a huge amount of information and can immediately tell you if you're dealing with something local to the user that reported the problem, or if it extends to their workplace, or even the wider world. There could, for example, be a regional or global outage with Office 365 or Amazon Web Services that's preventing people over vast geographic areas from getting any work done. The end user wouldn't know this, they'd just know they can't sign in or access their documents.

Once you know if the problem is local to the machine, local to the workplace, extends to the company regionally, in that country, or globally, or affects other companies and users in the region, country, or globally, you can begin the process of diagnosis.

IT Troubleshooting: The Movie

Once you have an idea of the scale of the problem you're facing, you'll know if you are operating locally to the device (such as the small-scale detective work you might find in a TV series), working much more broadly to find the problem (such as happens when the series is made into a movie), or working nationally or perhaps even internationally (as inevitably happens when the movie gets a sequel and they need to do things on a grander scale).

Know the scale of what you're dealing with, or what the maximum scale might be, can help considerably in your troubleshooting task. For example, you'll know there and then if you're likely to want to consult with other people or perhaps a service provider or software or hardware vendor to check the status of the problem and to search for a solution.

If you're able to pin the problem down to the smallest possible surface, it will help ease your own mind, as you all of sudden won't be faced with a vast issue, and it helps set absolutely boundaries. For example, you will know that for each strand of troubleshooting, you can take them so far, but that you wouldn't have to take them any further than a fixed point as there would be no benefit to doing so and no solution to be found past that fixed point.

The End of the Beginning

So now we've established the parameters for your fault and the maximum surface we're looking at. You'll find that on most occasions this will be a very small surface, and on others it will very quickly emerge that the vastness of the surface means it's an outage you just have to wait to be repaired.

If it's not something you just have to wait for though, you now need to continue joining the dots to find what actually *is* causing the problem.

What Are These Dots of Which You Speak?

So what are the dots you need to join, and how do they relate to software, hardware, services, and users? Well every problem will start with the user, so they'll almost certainly be included in your diagnosis somewhere. What were they doing at the time, what was the last thing they did, why were they doing it, and what were they doing at the time the murder was committed? Okay, so I made that last one up, but I'm sure you take my point that the kinds of questions you'd ask of the end users are very similar to the ones they'd be asked if they were shut in a darkened room with a police officer and a tape recorder.

It's always worth remembering that people won't give you all the information you ask for. This will be for several reasons. They might not consider it important (remember they aren't tech people), they might forget it because it's not the sort of thing they think their brain needs to retain, or they might simply overlook it. Thus the nature of the questions and the simplicity of the answers you seek will always be of the utmost importance.

Indeed, in Chapters 11 to 13, I'll show you how to put together high-quality paperwork that will make questioning end users simpler and less hit and miss process than it otherwise might be.

The next part of the puzzle is to obtain information from their computer on what has happened. I'll detail this in depth in Chapters 14 and 15, but on a desktop PC you have a *vast* amount of information available to you.

- **The Event Viewer** is the main port of call for a Windows PC. Nothing at all happens on a PC, from the opening of a program to a Blue Screen of Death without a report being made and stored. There is a huge amount of technical detail also provided, and I'll show you how to harness the full power of the Event Viewer later in this book. Additionally, the Event Viewer can create alerts when a specific problem or error recurs.
- **The Reliability Monitor** can give you concise information from the Event Viewer, showing when over a period of time crashes, errors, warnings, and critical events have occurred, and what they were.

- **The Task Manager** and **The Resource Monitor** present live information on every metric of computer use, from open network ports to running services. The Resource Monitor is also highly configurable, allowing you to drill down into live reports containing only the information you need to focus on.
- **Microsoft SysInternals Suite** is an optional download for Windows that contains tools allowing you to view and manage all manner of Windows processes, check open or locked files, and manage every type of startup item from a file association to a media codec.

When it comes to hardware, the dots become points on a chain. This goes back to the old biology song “The leg bone’s connected to the hip bone.” There are always points in the chain that you can trace, to give a few examples:

- **Printers** have a chain that can involve the driver, the port on the PC the printer is plugged into, the cable to the printer, the Wi-Fi or Bluetooth connection to the printer (and any associated hardware such as routers or switches), any Ethernet socket connected along the way, the power lead for the printer, the electrical socket and the power supply, and any physical switches either on the printer that control power or that may exist on doors to detect if they’re open.
- **A PC with no picture on startup** can involve the power supply for the PC, and the power lead for the monitor, along with any electrical sockets for both and the power supply, the graphics sockets on the PC and monitor and the cable connecting the two, the BIOS or EUFI settings for the monitor on the PC, and more besides.
- **A PC that will not start** can involve the power supply in the PC, the electrical socket and electrical supply, the motherboard in the PC, the memory or processor if the motherboard has no or has a disabled audio speaker for alerts, and more besides.
- **Internet or network connections** can involve everything from the network driver to any installed network services, the network socket on the PC, any ethernet cable, ethernet socket it’s plugged into,

the Wi-Fi hardware on the PC, the switch, router, other connection hardware, and perhaps even the local Cellular mast or telephone exchange, not to mention any online service providers such as Office 365 or Microsoft Azure.

Keeping an Open Mind

In my own experience people who are diagnosing a problem with a PC tend to fall into one of two camps for their initial diagnosis. There are people who will always first look to a driver or software issue and other people who will always first look to a hardware issue.

While both approaches have their merits, and they're more than often brought about by the personal experiences people have with troubleshooting, they're not really the best way to approach an IT troubleshooting problem.

I will always encourage people to keep an open mind when troubleshooting, and to start at the end and work backward to the beginning. If anybody approaches a problem assuming that it's most likely going to be a driver, or some other type of software issue, then there's a real danger of either overlooking something significant or of completely passing over an important diagnostic step in the chain.

Whatever problem you face in troubleshooting, it is incredibly rare to ever face the same problem twice. I hope that I've demonstrated that there are tens, or perhaps in some cases even hundreds of possible places where things might have broken down. Understanding if the path to that breakdown is short or long is the first step along the way to a successful diagnosis, and it's a very significant one.

Then being able to follow the breadcrumbs left for you and to join the dots together until you eventually see the big picture is utterly essential, and so an open mind is also essential at every stage.

You might even find that you hit a stumbling block and can't, as the saying goes, see the wood for the trees. On these occasions, bringing in a fresh pair of eyes from a colleague can sometimes illuminate the one thing you yourself have completely missed.

This brings me to probably the most important step in joining the dots together, and that is never to overtax yourself. It's crucial to know when you're getting tired and to recognize that a break from the problem is needed. If your mind is tired, then you will inevitably miss things, and whole sections of the chain can be overlooked.

Summary

IT troubleshooting, as I've said several times, is a holistic process, and you have to approach it in a holistic way. Anything is possible, nothing is impossible, and some aspects of IT can now have so many different components of both software and hardware along the chain, that it's sometimes impossible to correctly gauge if you've not missed something entirely.

In Chapters [11](#) to [13](#), I'll show you how to mitigate some of this by creating quality paperwork and reporting procedures that can help ensure that nothing is ever missed. In the next chapters though we'll look at the technical structure of IT systems, the problems associated with Human beings, and how we put IT and psychology together to take our diagnoses forward.

PART III

Understanding IT System Problems

CHAPTER 7

How IT Systems Are Structured

As you might have guessed from the first part of this book, IT support is a mixture of detective work, diplomacy, technical prowess, and sifting through masses of data on a regular basis. Well what if I told you that the IT support you provide can also be affected, or problems created by the natural or built environments in which we live and work?

You probably think I'm mad at this point for suggesting that trees, rivers, and skyscrapers can cause problems with computers, and you'd be right. I've never seen a single instance of a computer being adversely affected by a giant oak tree, or disrupted by a horse, but there are instances where our environment needs to be considered carefully.

We'll begin though by looking at our IT systems themselves. How do we structure and build them, what do we expect from them, and we'll look in a little more depth at the interconnectedness of things that I've mentioned before.

In the Beginning, the Unix-verse Was Created ...

What was your first computer? Mine was a Sinclair ZX81, a small black box about the size of a paperback book that plugged into my television set and a cassette player (remember those?). You could use two (read it, two) accessories including a RAM expansion pack and a thermal printer (which was utterly epic and once used to print my sister's wedding invitations).

By the time I'd moved to a PC, things hadn't changed very much. My Olivetti M240 had a monitor and keyboard (no mouse as I used MS DOS and DR DOS at the time) and had a dot matrix printer attached. Indeed I still have some printouts created using WordPerfect 5.1 on that machine.

Now things are very different. As I write this, the peripherals I have connected to my PC include an LG 34-inch ultra-wide monitor, Logitech wired keyboard and wireless mouse, Saitek flight controllers set (for playing *Elite: Dangerous*), an Elgato Stream Deck for additional gaming controls, an Xbox USB controller, a Bose volume knob that sits between the PC and my (also Bose) speakers, and an Asus external Wi-Fi antenna. The printer/scanner this time is connected over Wi-Fi.

If I look at the other connected devices on my home office network, there's the broadband modem (obviously); a Netgear Mesh router system with two satellites; NAS drive; IP security camera with base station and four wireless cameras and three Harmon Kardon Invoke Cortana speakers; my housemate's gaming PC, with connected monitor, mouse, and keyboard; My Dell XPS 13 laptop; three Gemini PDAs; my housemate's Samsung Galaxy S8 smartphone; an HP 18-inch tablet I use for watching TV in my office; the televisions both in the living room and my housemate's bedroom; the smart mattress cover on my bed (yes, really); and probably a few other things that have become so ubiquitous that I've forgotten about them.

Into this I'm planning to add a smart thermostat and intelligent lighting, not to mention an outdoor mesh router extender.

Let them look at the way IT systems are structured in a typical workplace. You will have multiple, sometimes hundreds of desktop and laptop PCs, just about everybody's smartphone will be connected to the company Wi-Fi, and likely a few tablets as well. There will be multiple printers, fax machines (possibly), and scanners of various description. There might be a local file server or NAS drive (or several), and perhaps a local server as well. There'll be a router, switch and patch boxes, and all the cabling to connect those switch boxes to ethernet sockets.

All of this equipment is essential to the smooth and daily running of the workplace (except perhaps the smart mattress cover), and they're all interconnected. It's this interconnectedness though that can prove to be the Achilles' heel of the system that you have.

Let's take the example of a large office where there are networking problems. Could this be caused by the switch box? Could it be caused by a faulty ethernet socket? Could it be caused by damaged or snagged cabling? Could it be the fault of the person who bought unshielded network cable that's now suffering from electrical interference? In fact it could be caused by *any* of these things and about another hundred things as well.

IP Freely

No, this isn't a joke from the Simpsons but important information about limitations of the current technologies we use. While IPv6 networking has been with us for some years now, it's still the older IPv4 networking system that dominates. This provides a physical limit of around 4.3 billion addresses, and we've long since outstripped that number for Internet- and network-connected devices worldwide.

A typical home router will allow for 253 devices to be connected if you are using a DHCP (Dynamic Host Control Protocol) server. Company networks use switch boxes and software-based address resolution trickery to extend this number as just having 253 devices able to connect to an office network wouldn't work in any workplace larger than about 30 people. You would very quickly run out.

With IPv6 there is a theoretical maximum of 340,282,366,920,938,463,463,374,607,431,768,211,456 possible addresses available, though the actual number is more like a smaller 42 undecillion, not 340 undecillion. Even so this is more addresses than we'll need for a significant amount of time.

Most network systems still use the older IPv4 system however, and it's much easier to access a router admin system by typing 192.168.1.1 than it is to remember and type (as an example) 0000:0000:0000:0000:0000:FFFF:C0A8:0001.

Older tech doesn't just stop at our home and company networks, the whole Internet runs on it. Both the Internet and the World Wide Web (they're actually different things with the latter working on top of the former) can be dated back decades. The World Wide Web was invented by Tim Berners-Lee at CERN in 1989, while the Internet can be dated back to the US Military's ARPANET system in the 1960s.

The end result, and I'm sure I don't need to tell you this, is that neither is properly equipped for the uses we put them today. The lack of base security and authentication have permitted spam and phishing email and criminal activity to run rampant for years. On the flip side, they have also facilitated private communications for people in oppressive regimes, but the dark web with its sales of drugs, weapons, and stolen personal data is a direct consequence of the open nature of the Internet and the World Wide Web.

Aging Tech

In these days of smartphones where people upgrade their handset every year or two, it's easy to believe that technology moves at an incredible pace and that everything we use is "the latest thing." I've already demonstrated however as in the cases of IPv4 and the Internet that this simply isn't true. There was a plan a few years back to roll out "The Internet v2.0" which would have had personal security certification baked-in. This technology would have shut down the online market for spam and phishing emails overnight as it would have made it incredibly simple to identify where an email had originated from and who it was that owned the account.

As brilliant as this would be, it hit two major snags which ultimately stopped it dead in its tracks. The first was human rights organizations such as Amnesty International, who were concerned this new tech would enable oppressive regimes to easily and immediately identify anybody online who dissented from established government rules.

The second problem though was bigger, much bigger. This was the Internet itself and the issue of how you migrate such a massive infrastructure of web sites, servers, and online services over a new system, or whether you have to run the two systems side-by-side while people and companies transition from one system to another. Simply running the new system on top of the existing one wouldn't be wholly practicable for reasons I shall come to.

We've all seen examples of where business especially doesn't want to move to new technologies, and indeed stubbornly refuse to do so, and I'll deal with probably the most high-profile example shortly. It was very likely then that the lofty goal of abolishing the existing Internet infrastructure might never be achieved, and in the interim all that would happen is create two competing and entirely separate infrastructures that would confuse consumers, cut businesses off from one another, and ultimately cost a vast amount of money that no company, organization, or government was prepared to pay for.

So we end up with the fudge of new technologies being built on top of the existing Internet infrastructure that in part, as I mentioned before, dates back to the 1960s. This infrastructure, on which runs Sir Tim Berners-Lee's World Wide Web, was never designed for the modern Internet and as such contains flaws that are exploited by criminals and hackers alike. Building Internet v2 on the top of this would probably not alleviate these problems.

Windows NT

The first versions of Microsoft Windows were built on top of Microsoft DOS (Disk Operating System), and this remained the case until long after Windows 95 and the introduction of what we now consider the modern Windows UI (User Interface), which was still a graphical user interface (GUI) that sat on top of the DOS core. Microsoft was very clearly falling behind with companies such as IBM and Apple nipping at their heels and developing full GUI operating systems such as MacOS (1984) and OS/2 Warp (1987) on which Microsoft itself worked and contributed. Microsoft was falling badly behind and needed a secure, stable, and reliable version of Windows on which to build its business customer base. The end result was Windows NT (New Technology) which was released in 1993, shortly after Microsoft severed its partnership with IBM, taking its copy of the OS/2 codebase with it.

Windows NT remained a resolutely business-only operating system until 2001, when the DOS-based consumer operating systems, the last of which was the horribly maligned Windows Me (Millennium Edition), were finally decommissioned. The advantages that NT had however were that it was designed from the ground up to be a full GUI OS, and while DOS remained for compatibility and scripting purposes, it was no longer the operating system itself.

This allowed for Windows NT to be more secure, more stable, and more fully featured, especially considering that MS DOS was first launched in 1981 and therefore wasn't equipped to support many technologies that came along afterward, such as USB (Universal Serial Bus), Plug and Play, or UEFI (Unified Extensible Firmware Interface).

However, it can be argued, and I would certainly make this case, that Microsoft made one pretty huge mistake with NT. This being that they allowed, and still allow to this day, compatibility with software and hardware that ran on DOS. This means that all of the underlying code, features, and security in Windows 10 (the latest version of the OS) have to be compatible with everything going back to Windows 95; Windows 1 through 3.11 had a very different core architecture that was dumped when Windows 95 was released.

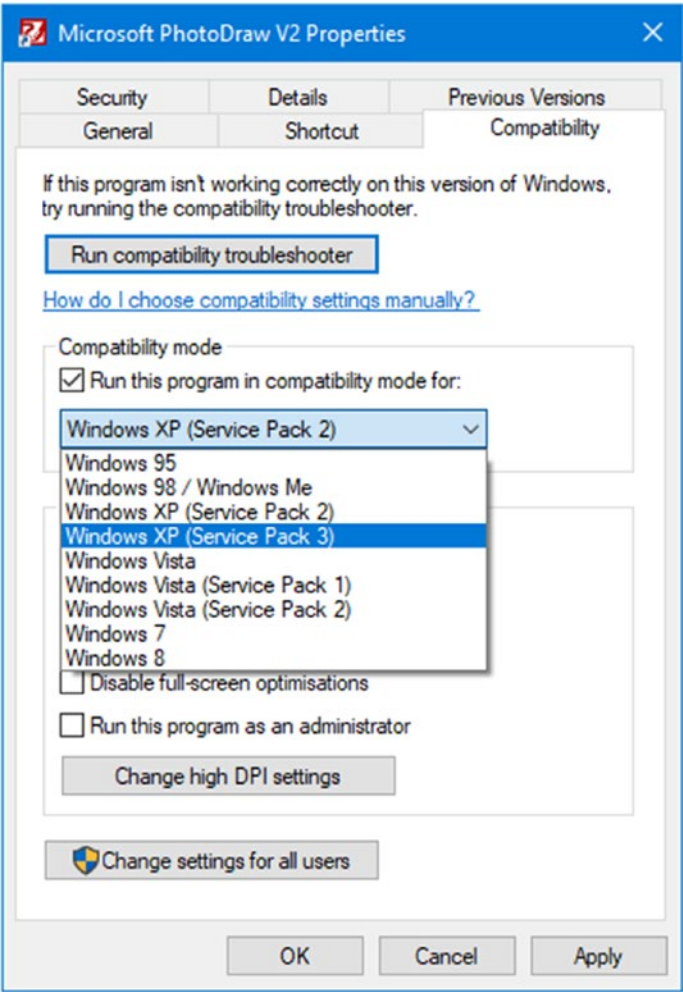


Figure 7-1. The Windows Compatibility Settings

Even the Windows 10 Device Manager allows you to *Install legacy hardware* that includes devices running on Serial (RS232) and Parallel interfaces, or even via infrared. Engineers love this as it allows their PCs to interface with truly analogue testing equipment as Serial and Parallel were analogue technologies (transmitting electrical waveforms at various baud rates) rather than digital which wasn't properly adopted until the advent of USB in 1996. If you ever see a laptop that still includes a Serial port, put it on eBay and some engineer will almost certainly buy it from you.

The upshot of all this, and I know you’ve been itching for me to get to the point, is it created a disincentive for businesses to update their software or hardware every few years when a new version of Windows was released. Various attempts were made by Microsoft to beef up the operating system, such as the User Account Control (UAC) security feature introduced with Windows Vista in 2006, but businesses still stubbornly refused to update their software and hardware.

This problem was exacerbated further by the end of support for Windows XP, which many people felt was as comfortable as an old shoe, and also for Windows 7 for the same reason. This was a significant driver in Microsoft’s decision to declare that Windows 10 would be the last ever major version of the operating system, and it would then get annual (or semi-annual) iterative updates to help reduce costs for businesses and in their move to drive *Windows as a Service* (WaaS) rather than as a product.

Businesses didn’t like migrating to a new version of Windows every 3 years, nor did they appreciate the costs involved. For this reason, many businesses skipped whole Windows versions, skipping Vista to move from XP to Windows 7, and skipping Windows 8 and 8.1 to move from Windows 7 to Windows 10.¹

Windows vNext

Microsoft has had several attempts over the years to produce what’s often called Windows vNext (version Next), and this has come in several forms from an entirely new managed code operating system that wasn’t compatible with Windows codenamed Midori, to a virtualized OS in which all software, hardware, and services are run in segregated virtual machines.

We can see some of this technology today with Windows on ARM (WoA). Microsoft has been pushing software developers, with very limited success, toward their Universal Windows app Platform (UWP), even providing a service called centennial bridge, that allowed “traditional” win32 desktop software to be repackaged as store apps. These apps all run in a virtualized container in Windows 10, making them both more stable and more secure.

¹It used to be said that Windows versions were like classic *Star Trek* movies, with each other one being terrible. The common offenders often being cited as *Star Trek: The Motion Picture*, *Star Trek III: The Search for Spock*, and *Star Trek V: The Final Frontier*. Similarly, Windows Vista and Windows 8 are almost universally derided.

Windows on Arm takes this technology one step further by providing in-built support for virtualized win32 apps. At the time of writing, not very many win32 desktop programs are supported, and as a result the operating system hasn't been a huge success ... yet. Fortunately the primary advantage of the ARM processor is efficiency and the first WoA laptops and 2-in-1's sported battery life of up to 20 hours, so there's still hope for the platform yet.

Microsoft has proven highly effective at creating virtualized operating systems and app platforms; their App-V technology for business app virtualization was first introduced in 2006. It's still possible then that a new version of Windows will eventually land and that Windows 10 won't be the last version after all. It makes complete sense both for software and hardware compatibility, security, and stability, to have absolutely everything running in virtualized containers, and modern PC hardware is certainly capable of running such an operating system.

The problem is the Windows user base, and there are currently about 1.5 billion PCs in use worldwide. That's a *lot* of people, businesses, organizations, and governments to encourage to move to a new platform. Given the trouble Microsoft had just getting people to move away from Windows XP it's very likely that we won't see this happen for at least another decade or two.

Creating a New Android

It's not just Microsoft facing this problem. It's rumoured that Google are also developing a next-generation version of Android to build on what they've learned from the development of the OS from its first release in 2008 to now.

Though not much is known about the product, it's very possible it could be incompatible with the existing Android OS core, meaning apps would need to be re-written or run in virtualized containers.

Apple has the problem that their core desktop operating system is now also very old. Having first been released in 2001, it is definitely beginning to show signs of strain. Fortunately for Apple (depending on how you view these things), they have a much smaller and much more loyal user base than either Microsoft or Google, so should a dramatic shift occur on the platform, it would be less likely to upset the masses.

The Upshot

The upshot of all of this is that the modern computer age finds us supporting a wider range of software and hardware than could ever have been predicted, with the resultant problems that mixing the old and the new often brings.

If we only had to deal with containered apps and device drivers, then we'd see almost no real problems on our computers that couldn't be resolved by uninstalling and reinstalling the corrupt component. Instead we have core OS files that are still accessible to file managers (in Windows, Android, and OS X), apps that are critical for business use, or comfortable for the end user and that are more than 20 years old, and hardware that is still essential because businesses, organizations, and governments stubbornly refuse to update the software that uses them, as they have little incentive to do so as long as everything "just works."

This refusal to update operating systems and software has the knock-on effect that our entire network and Internet infrastructures are also held back and are now showing signs of creaking themselves. It's all a bit of a mess for the IT support technician, and it's not going to get any better, any time soon.

Living in the Internet Age

We might look back nostalgically to the computers we used to own and use, and this is partly because of the simplicity of the times. Simpler times before social media dominated everybody's lives, spam and phishing emails dropped in your inbox every day, and viruses potentially lay in every document you were sent.

The Internet does bring huge advantages, while the average person on the street might be grateful for how convenient their shopping and banking has now become, but there's more to it than that. The Internet has allowed dissidents in oppressive regimes to communicate with one another, people in isolated parts of the world, and people who might otherwise be isolated within their own communities to stay in touch with family, and to expand their network of friends.

It's this very interconnectedness that now present both the biggest opportunity and the biggest headache for people providing IT support.

Oh My God! The World Just Ended!

If you've ever been in the presence of somebody who had just lost their Internet connection, you might believe that the world had just ended. Without the ability to check their email, messaging, and social media, without being able to shop online or check their credit card statement, and without (with horror) the ability to search for good places for next year's summer holiday, it seems that people suddenly revert to being Neanderthals, unable to even feed themselves properly, let alone function in the real world; actually this is unfair to Neanderthals.

This means that the moment somebody's computer is disconnected from the Internet, their world has effectively ended, and they will constantly pester you, looking over your shoulder and asking "is it working yet?" until they can once again get back online and do pretty much the same absolutely nothing they were doing before.

Human beings are a pain in the butt and should be strongly encouraged at every turn to use that mythical thing called a front door that will lead them into a new and exciting world of adventure and exploration. This does however highlight a fundamental problem that we have these days with our computers.

The Nearness of You(Tube)

How do you properly, quickly, and easily diagnose problems with a PC when it's simultaneously connected to almost every other computer on the planet? I'm not exaggerating here either, as scenarios such as distributed denial of service attacks (DDoS) and worldwide malware and ransomware attacks demonstrate just how interconnected everything is.

This is something you always have to consider, and it's why I encourage taking such a holistic approach to computer troubleshooting. Even a problem with a printer these days could be caused by an outage at the local ISP.

You might be surprised too at just how often this interconnectedness is actually the cause of a problem, when it appears that the problem is actually caused by something local on the PC. As an example you might find that there is an authentication problem with the Microsoft Office 365 servers that is preventing the user from being able to access their files, when the problem could appear to be a permissions issue with their account.

For this reason it's always wise to keep a few useful web links to hand. These links are able to tell you the status of a web service or site and inform you if it's working appropriately.

- www.isitdownrighnow.com
- www.downdetector.com
- portal.office.com/servicestatus
- www.google.com/appsstatus
- status.aws.amazon.com

These web sites can help you quickly determine if the fault is being caused by an outage, or fault with the service provider used by the customer.

There is one more link that I can highly recommend for determining an outage with an online service: www.twitter.com, as you can guarantee that *any* outage will be immediately reported, inflamed, and shouted at by people on social media (unless Twitter goes down that is), and you'll very often get a faster response on Twitter than from any company or third-party service status page.

Hardware Is Hard Wearing

Here we are, almost at the end of this chapter about how IT systems are structured, and I finally get around to mentioning hardware. There's a reason for this, being that hardware these days is pretty damn robust.

If you go back to the early days of computers, you had motherboards on which the silicon chips (as we so lovingly called them at the time) were removable, and where individual capacitors would sit upright from the motherboard. We've long since moved into a world of surface-mounted, non-serviceable components, where nothing is removable, and you'd be hard-pressed to even identify an individual component anyway as they're all so small.

This means if you get a hardware failure, it's more likely to cause the entire machine to fail, rather than cause a problem. This makes the process of diagnosing some issues simpler but can also make others more complicated.

Hardware Also Wears Out

Let's take the example of a desktop PC which will turn on briefly, and then shut itself off again. What is it that could be causing the problem?

There are actually several different components in the PC that are causing the problem. It could be the power supply that's getting old, and just like an automobile engine, not delivering as much power as it did when it was new. It could be a component on the motherboard that has failed, or it could be fans that regulate the heat in the PC (this one is easier to diagnose). You could have a problem with the power lead, the electricity supply, the surge protector, socket splitter, or uninterruptable power supply (UPS) the computer is plugged into, or even the physical power button on the computer getting stuck or having a broken micro-switch.

This is why I stress looking at the problem holistically, and first eliminating what's definitely **not** causing the problem; you will save a lot of time and effort this way.

Summary

So what is the lesson to take from all of this? Quite simply it's that the hardware we use is both the biggest headache we can face and also quite possibly the last cause of any problem. Much of the technology we use and rely on these days is so old that it could have been in use before you were born, and you might well wonder why we're still using it.

This mix of the old and the new comes with and causes its own unique problems, most notably with compatibility and getting everything to work and run properly together.

Speaking of being born and working happily together, this brings me neatly on to the next and definitely the most frightening piece of the IT support puzzle ... the human being and the problems they can cause.

CHAPTER 8

The Human Factor

How the Human Factor and Staff Training Affect IT Systems

People are a problem, a big problem. We've already established that nothing ever goes wrong with a computer without a Human being involved in some way. It can therefore be argued that these people simply should never be allowed to use computers anyway, as they just create work for IT pros.

That would be the circumstance in an ideal world anyway, but it would have the unhappy side effect of hampering productivity somewhat, we'd all have to go back to farming.

So why aren't people trained to use computers? Well the answer to this is complex and varies from one business or organization to another, but there is always training going on. Companies provide in-house training which I'll talk more about shortly, and there's a huge market in books and online courseware. This book that you're reading now is my 19th published book (and I'm very excited that I'll certainly get to 20). I also, at the time of publication here, have 14 video courses live on Pluralsight. Additionally, I provide bite-sized help, support, and training on my own web site <https://windows.do>.

You will also no doubt provide training to some degree. This might be just helping people to understand the problem that occurred on their PC, so they can either fix it themselves should it happen again or more likely avoid it altogether in the future. You could also provide training courses or courseware within your company or organization. Indeed it's very common for IT professionals to have to deliver training in some fashion, from inductions on how to use company software and systems to training days when new updates are rolled out.

You may see these as a chore, but they're really a valuable opportunity to help reduce your own workload if you do things right. You can use them to instill best practice in employees and management alike and to explain to them why it's not only important to

the company but important to the team and the efficiency of the people around them that certain things are done a certain way.

Back in Chapter 3 I wrote about learning theory and how you can structure staff and IT training so as to maximize the impact of the message you're trying to convey, while keeping the people you're teaching engaged, and making sure that everybody learns and comes away from the training having achieved something, both for themselves and for yourself.

You might not consider teaching an important skill in your job; it could just be an ongoing chore that keeps repetitively cropping up and interrupting valuable workflow. Teaching however is crucial and I can't stress this enough. If you get it right, then the rest of your workflow will become less hectic and less busy.

In order for training to work, however, you need to understand where the people who use your IT systems are coming from, what their needs are, and how they view and see the world. This brings me neatly onto the subject of...

Why Users Screw Up IT Systems

People always mess with their computers, be it the hardware, the software, or the settings, and they have a nasty habit of screwing things up. If you're to understand how to prevent these problems from occurring, and how you can effectively train people and instill best practice into them, it's first necessary to understand *why* they screw up IT systems and computers in the first place.

Even though there are often connections to be made between the different things that people screw around with, we'll deal with each of the three main subject areas separately, beginning with...

Hardware

People are horrible to hardware. As an example of this, I have seen so many iPhones with cracked screens over the years that after a while I almost began thinking they were shipped like that. So what are the reasons why people treat hardware so abusively? In part it comes down to something my father always told me as a child who sometimes described me as someone "who knows the price of everything and the value of nothing." This is an Oscar Wilde quote, and my father rather used it out of its original context. The basis of it stands true though with our tech.

Everybody knows how much their technology costs, and prices have been rising steadily in recent years. The cost of a high-end smartphone from the likes of Apple or Samsung is now well over \$1,000, and a high-end ultrabook can cost \$2,000. Everything is very expensive indeed.

This doesn't stop people dropping their phones on the pavement, however, or spilling their drink on their laptop, or even dropping their phone down the toilet. There are two reasons why these events happen.

The first is that people are completely addicted to these devices, and as a result they end up using them in the most unsafe environments, or at the most inappropriate times. Walking down the street while messaging people on social media is a perfect way to walk straight into a lamp post, and texting while driving (while illegal in many countries) also takes all your concentration away from the road.

The other reason though is why the hardware gets abused. It's a tool, nothing more. People might say they have an emotional attachment to their smartphone, but it's not the phone they love, it's what that phone does for them, and the wider world that it connects them too. If you took away that phone and replaced it with a different one that was configured to allow access to the same web sites, social media accounts, and games as the first phone, that person would adjust to the new device very quickly indeed and forget about the old one just as quickly.

This means that the physical hardware itself is of very little value to the end user, even if it cost them upward of \$1,000. They only learn this lesson (sometimes) when they cause it damage such as cracking its screen and then suddenly realize they can't afford the \$300 or more it will cost to have that screen replaced, thus the huge volume of smartphones with cracked screens still in use day to day.

When you give people hardware to use, be it a smartphone, a laptop, or something else, or when you're purchasing hardware for people to use, this is an important consideration, and it's one of the reasons that Lenovo laptops consistently prove so popular. Ruggedness and simplicity are built into Lenovo laptops by design. They're not flashy like the latest designs from HP, or stylish like ultrabooks from Dell, they're reliable, dependable, strong, and cheap to repair.

Smartphones are a different matter, and cause additional problems, not the least because people kick up a hell of a fuss if you give them an operating system they're not familiar with or that can't run the games they've bought (again this is an issue quickly resolved through training). It's very common these days for phones to come with Corning Gorilla glass screens and both dust and water resistance. These are

considerations when buying handsets, yes, but so is the cost of replacing the screen, or the battery. If a phone comes with the best and most shatter-proof glass on the market (and these things are *never* as good as the marketing might suggest) and can be used at the bottom of the Mariana Trench, they're still no good to you if it's gonna cost \$350 every time the screen cracks.

I know it's unpopular, especially with executives, and it's the IT equivalent of making the staff drink their coffee from a beaker, but sometimes these people just have to suck it up and accept that if they can't be trusted to look after their toys, they can't have the nice things. You've just gotta be tough on them as, honestly, it's the only way people learn.

Software

To a large extent, the problems of software meddling can be addressed using user and security policies. If you have ever run a network through that was locked down using group policy, there's still always people who manage to subvert this in some way to install their favorite piece of software, or to update something that shouldn't be updated, or to remove something that shouldn't be removed.

This is much harder with mobile devices such as smartphones and Chromebooks too, as people *expect* to be able to install any old crap from the store, and can get very obnoxious when you try to prevent it. One carefully worded email from an employee to their senior manager about how it's essential they're able to play Candy Crush because they travel a lot for work and are a nervous flyer can sometimes be enough to bring months of careful security planning crashing down around your ears.

People tinker with software though, install new software, and uninstall existing software because they get bored or frustrated. Somebody who is genuinely involved in what they're doing would have neither the time nor the interest in updating their software or installing a new piece of software.

It might not seem like it, but this is sometimes something you can do about as an IT professional. It could be, for example, they find the current software just too difficult or frustrating to use, meaning either the software could do with an update or a usability overhaul or that the training people receive in how to use the software isn't as great as it otherwise could be.

Settings

With Settings it's often a similar reason as it is with software, frustration or boredom. You might be surprised at just how many people *never* change the default settings on their smartphone, tablet, laptop, or desktop PC. You might also be surprised by the sheer volume of people who don't even know that the option to configure settings such as the look and feel of a device even exists.

People therefore only play with settings when they're bored or frustrated. Again though, as with software, this frustration could be caused by a lack of understanding or training or through poor usability.

There is another factor though that comes into play here, accessibility. It's important to remember that the Human condition is almost infinite. People have different genetic makeups, different backgrounds, different likes, and problems that they face, and some have medical or genetic issues that can cause barriers for them when using technology.

IT and Accessibility

In my Apress book *The Windows 10 Accessibility Handbook*, I make the point that accessibility for our PCs isn't just something to be used by those with special visual, auditory, motor, or cognitive needs, but how they can help people with everything from shaky hands to color blindness, dyslexia, and dyscalculia. It can help people who are older, or who have difficulty concentrating; it can help people who work in noisy or distractive environments and those suffering from stress or mental health issues.

It's not just about the accessibility features built into the operating system either, and iOS, Android, OS X, and Windows all include many of them. Features such as night light, which reduces the amount of blue light emitted by the screen, can help people who work late at night to concentrate, relax, and rest. The desktop scaling settings can help people whose eyesight is less than 20-20. The ability to add visual notifications to sounds can help people who work in warehouses and factories.

It's well worth your time as an IT professional, understanding what accessibility and ease of access features are available on the devices you use, and spending some time researching *how* they can be used. With the right help and support, you can really help the individual and, in turn, the team, the company, and yourself.

Users Are Not IT People

Would you describe IT pros as “normal people”? It’s very common when we talk about our own lives that we refer to non-technical individuals as normal, because that’s what they are. Everybody has things that they’re good at, some people are skilled chefs, some are horticulturalists, some people understand mechanics or electronics, and some people have specialist skills such as being able to repair damage to the Human nervous system.

It’s no different with IT. What you do is a specialist job; it’s something you have to train for, for years, in order to become proficient at, no matter how much raw talent you might have in the beginning. Normal people don’t have these skills, nor do they want them.

Earlier in this chapter, I said that people view their devices as *tools*, and this is because that’s what they both want and need them to be. They need to get stuff done, write a letter, submit a report, calculate a ledger, manage finances and investments, or buy a special present for the new baby coming to the family. They aren’t interested in how their IT works, just that it does.

Sometimes it can be very difficult to look at things from other people’s points of view, but when it comes to providing IT support, it’s absolutely essential. Not only does it help you understand what’s happened, but it also helps the person you’re helping understand how they can avoid the problem recurring. Empathy and understanding are essential tools in every IT pros toolkit.

The Monkey Mind

The Monkey Mind (sometimes called the Mind Monkey) is a Buddhist term that describes those who are unsettled, restless, capricious, whimsical, fanciful, inconstant, confused, indecisive, uncontrollable, and who use computers (okay, I made that last one up). It’s a great description though of what happens to each and every one of us through each day.

It’s far too easy, and frankly conceitful, to assume that people come to work, focus completely on their job, get done what they need to do efficiently and to the very best of their ability, and then completely switch off from work the moment they leave the workplace. This is wrong because it’s just completely impossible.

Every single day each person will think tens of thousands of different things. They'll think about the problems they faced getting to work, about the shopping they need or want to do, the school play coming up for their young child, the anniversary they keep putting off preparing for, the holiday they want to take later in the year, the difficult relationship they're having, the wedding that seems to occupy every moment of every day, the car repairs that somehow need to be paid for, the mortgage that's just gone overdue, the credit card that's maxed out ... again, the football practice for the kids they have to rush home for, the medical condition that's given them a blinding headache, and so on.

It's very easy then to see why problems occur with our IT systems. This is because people almost never give their full concentration to anything. As an example, while I'm writing this, I'm listening to the Jean-Michel Jarre album *Chronology*, bemoaning the lingering cough I've had sitting on my chest for the last 4 weeks, and thinking about when I'm going to find the time to box up a couple of packages that need to go in the post tomorrow.

It's very difficult for anybody to full concentrate, and therefore missing something, getting something wrong, or perhaps even being tricked into doing the wrong thing by a piece of malware or a scammer is all too easy.

There's nothing wrong with any of this, it's just Human nature, but it's yet another consideration for IT pros who have to diagnose and repair the problems caused by others. Pete in accounts was planning his wedding when the order came through for toner cartridges for the fourth floor. Several days later when you get the call to say the printer's broken, it turns out that Pete typed the wrong digit for the toner product code and ordered the wrong one.

Mary has moved from her desk at the back of the office to a nicer one close to the window. It took her an hour, but she's been able to move her PC, monitor, and all her files, accessories, and plants from one desk to another, but now her PC doesn't work. It's not for Mary to think that there's two networking rings and she's plugged her PC into the wrong one, because she's got a sick 7-year-old who's been sent home from school and is being looked after by his grandmother.

People Are Complex

Human beings are the most complex life form on the planet. If you ever looked at a cat, or any other animal, bird, fish, or mammal going about its everyday business, and wondered for a moment how much simpler life must be for them, it's because they lack the complexity of the Human condition.

This Human condition is what has broken the speed of sound, put people on the moon, it's what has created the greatest engineering, and the most beautiful art. Sadly it's also what causes arguments, insecurities, insincerity, hate, violence, war, and envy.

For as long as we understand this, we'll always be able to provide quality IT support to people and IT systems for them that they can use and understand. We'll be able to provide training that works, because we've not tried to throw too much at them too quickly. We'll be able to have people help themselves, because we helped them understand what happened and why.

Every great achievement in Human history is because our species has an insatiable desire to learn new things, and a fantastic endorphin response from the brain when it happens. That little "I did good" kick that we get when we realized that we have learned something is what drives us and spurs us to do new, bigger, and better things.

You can take full advantage of this and harness it. You don't need to be a psychologist, or a behavioral expert to know how either. Taking baby steps in helping people join the dots from A to B and seeing the reaction you get when the penny drops can give you the same endorphin hit the other person or people have just had.

Training and understanding are something we all desire, no matter how much work might be get done. Sometimes the initial barriers might seem a little high, but with time, practice, and a little patience, we can help tame the human condition and help everybody achieve a little more, a little better.

Summary

It's so common to think of IT problems as being caused by IT systems, both software and hardware, that sometimes it's all too easy to overlook how complex the Human factor might be. People don't *want* to break things, but there's always so much going on inside the Human mind that even the simplest task such as plugging in a network cable can go wrong.

Which brings me neatly onto my next subject, where in Chapter 9, we'll look at peripherals, how they can very often be a thorn in the side of any IT professional, and what you can do to mitigate some of the common problems that you face.

CHAPTER 9

The Peripheral Problem

We all use peripherals with our computers, even if it's just a keyboard, mouse, and printer. You might be surprised however at the actual range and scope of peripherals in use today. The peripherals that people use are also many and varied and include everything from specialist engineering and monitoring equipment to musical instruments, machinery, and virtual reality headsets.

Sometimes it can be all too easy to think of peripherals as just a printer, games controller, or a USB-powered desk fan. These many and varied peripheral types though can present real challenges in troubleshooting and repairing problems when they occur.

Riding the Legacy Wave

One such problem is that of legacy devices. There are a great many legacy peripherals still being manufactured today, and it's still possible to find brand new laptops sporting RS232 Serial ports. These laptops and equipment are almost always used by the engineering and scientific communities who need the accuracy that analogue devices offer.

If you have a sensor of some type, then the analogue waveform you get from its readings can be significantly more accurate than a digital device. This is because all digital sensors can only measure things in "steps." An analogue wave gives you far more nuance in the results; see Figure 9-1.

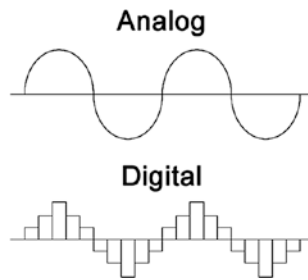


Figure 9-1. *Analogue devices can offer much greater resolution for sensors than digital devices*

This means that analogue devices can never be attached to a computer via a digital interface such as USB, as doing so would force the conversion of the signal data into a digital format. Older (legacy) connection standards though are serial in nature, such as the RS232 (D-Type) port.

Adding Legacy Devices to Windows

Legacy devices don't have a plug-and-play interface, which means that when you attach them to a computer, they're not automatically detected and configured. A legacy device is one that uses an older connection type, such as a serial or parallel socket, and will often need configuring or tweaking manually in order for it to work.

You add legacy devices in Windows through the Device Manager. Open the *Action* menu, and select *Add legacy hardware* from the drop-down menu; see Figure 9-2.

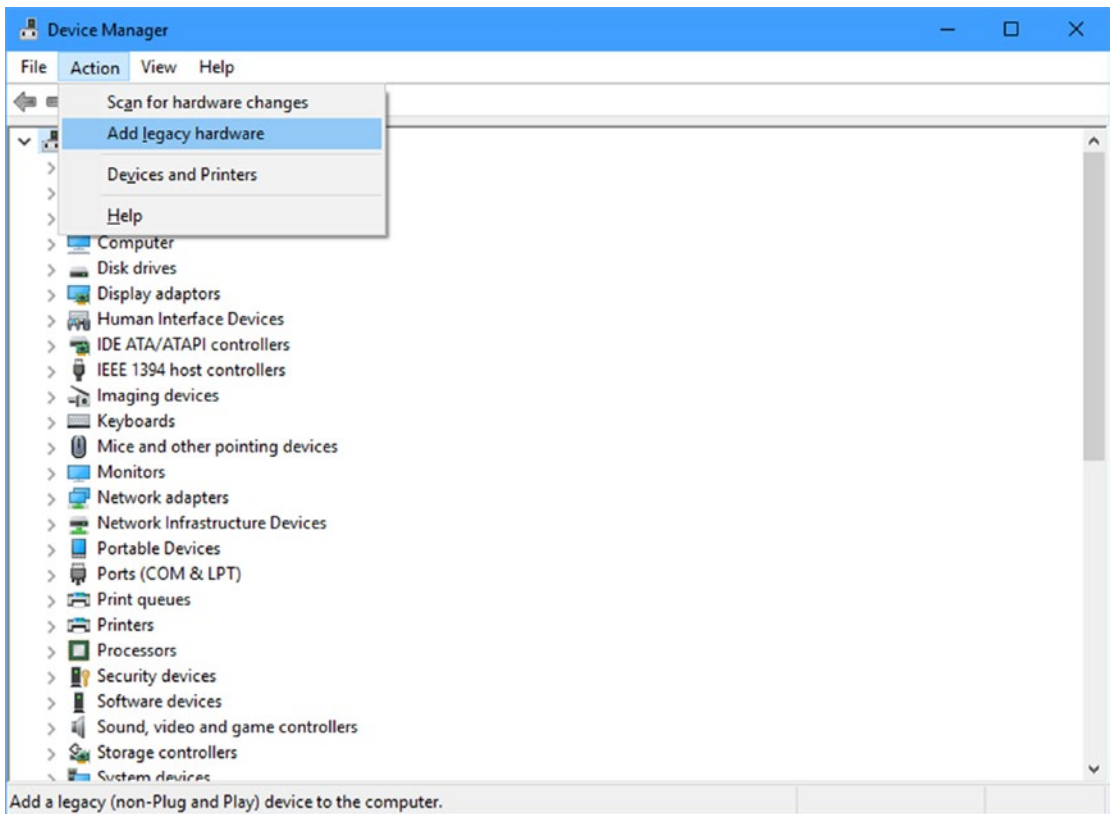


Figure 9-2. *You add legacy devices in Device Manager*

The legacy hardware wizard guides you through different types of devices you can use with a PC, including modems, PCMCIA adapters, and scanners; see Figure 9-3. Choosing a device type in this way allows Windows to perform some automatic configuration for the device to help you get it working.

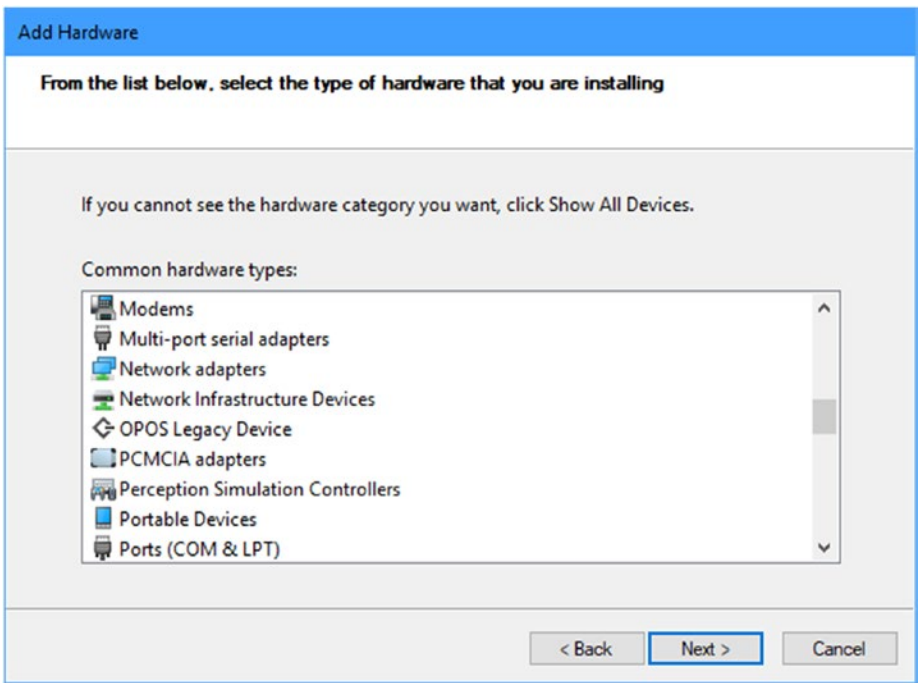


Figure 9-3. *You can choose the type of device*

It’s also in this hardware type list that you will find Serial and Parallel ports if they’re not already installed in Device Manager and available for use.

Another advantage of choosing the device type is that Windows comes with a wide range of legacy drivers available for installation, and you’re asked if you want to install one of the drivers Windows already has, or if the specific device isn’t listed, install your own driver from disk; see Figure 9-4.

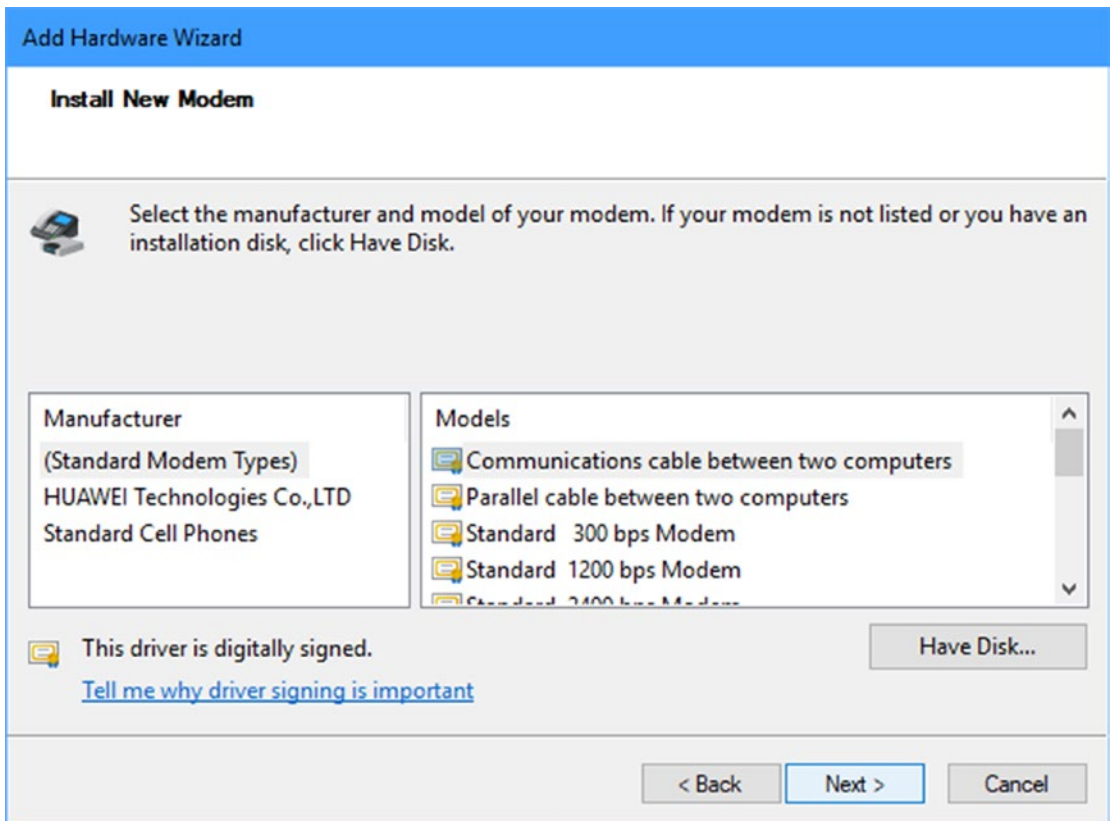


Figure 9-4. Windows comes with a wide selection of legacy drivers

Configuring and Troubleshooting Legacy Devices

Once a legacy device is installed, you will probably find that it doesn't work perfectly and properly first time and will need additional configuration. How you configure the device will be determined by the instructions provided by the device's manufacturer, and you should check their documentation.

If you right-click on a device in Device Manager, and then from the context menu that appears click *Properties*, you will see all the available configuration options for the device and the driver.

These will vary considerably from one device to another; indeed two different devices of the same type (e.g., two printers) will very often have different available options from one another. On the whole though the options you see will be straightforward and sensibly arranged; see Figure 9-5.

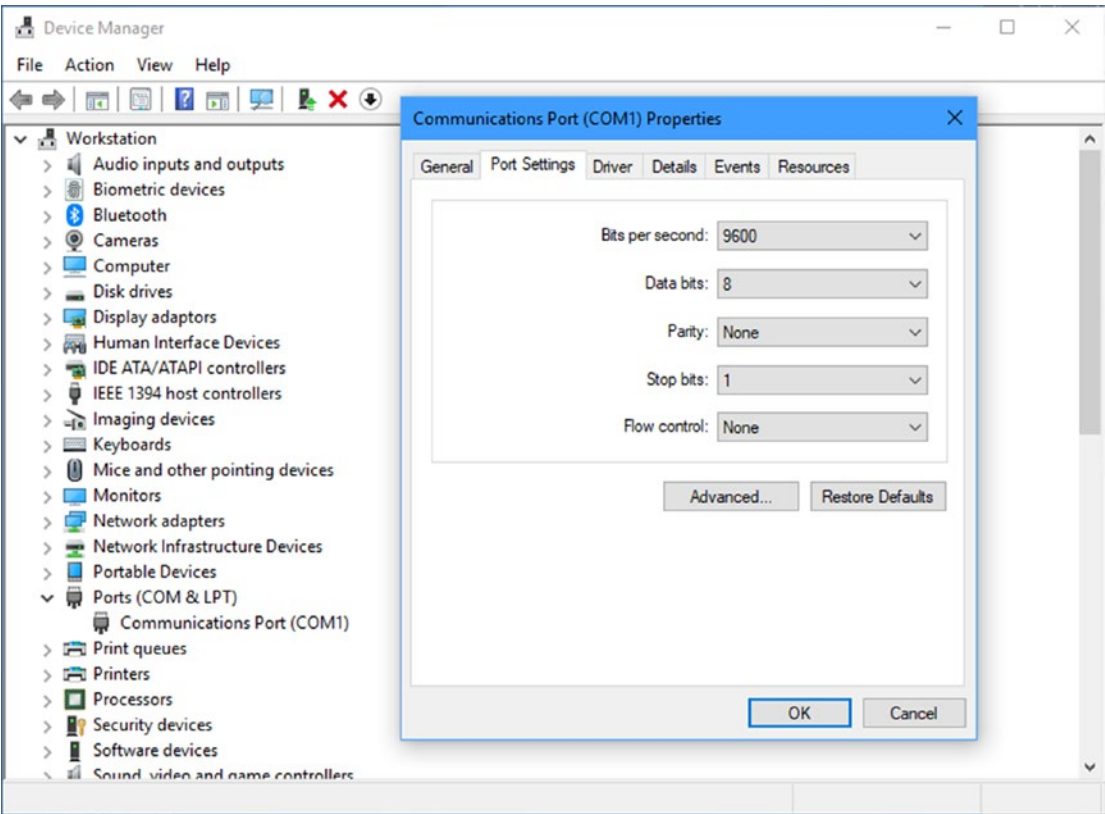


Figure 9-5. You configure drivers with Device Manager

Troubleshooting Device Drivers

When it comes to troubleshooting problems with device drivers, there are several ways to go about the process. I will start this by talking about System Restore, which you can find by searching for it in the Start Menu. System Restore is most useful if a change has recently been made that has caused something to malfunction, or misbehave, and it's most likely a Windows or other update that causes the problem.

You can use System Restore to roll back any changes and updates so that you can get devices working again. Those updates and patches will still most likely need to be applied, but doing so manually might solve your problem.

Otherwise, you troubleshoot and repair drivers in their *Properties* panel. You will see in Figure 9-6 I have two different types of driver, a legacy modem and a graphics card.

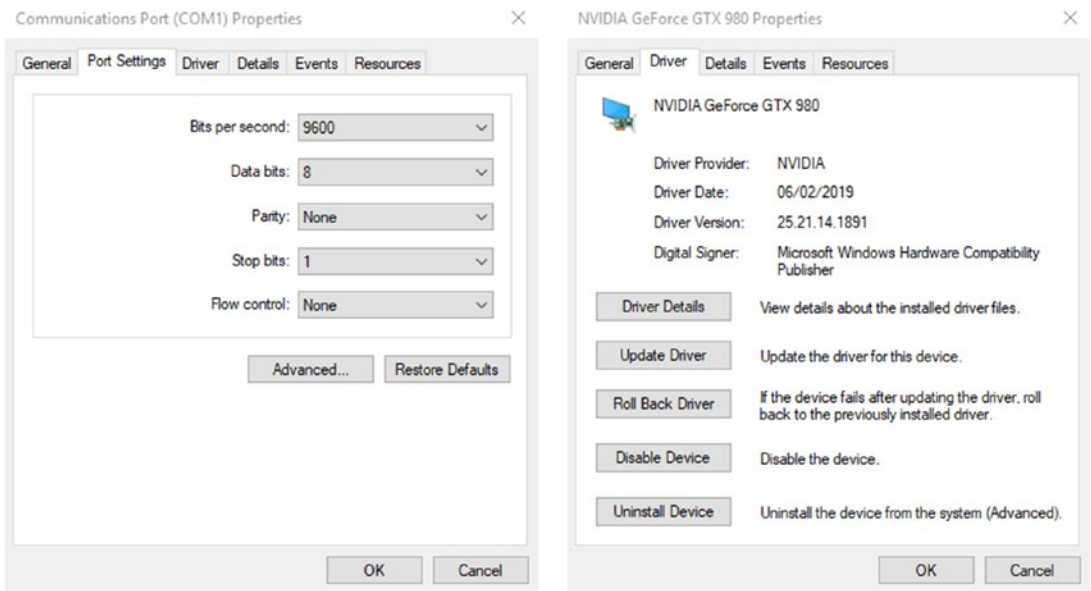


Figure 9-6. *There are different ways to troubleshoot drivers*

I wanted to highlight the legacy driver as in this case, it won't always be the case with all legacy drivers, there is a *Restore defaults* button. This will reset the driver configuration to its default state. If something goes wrong with a driver's configuration, it can sometimes be much easier to reset it completely and reconfigure it from scratch than it is to try and find the settings or settings that are misconfigured.

Alternatively, under the *Driver* tab, there is a *Roll back driver button*. This is only available if the driver has been updated at some point, such as an update being delivered through Windows Update. As an alternative to using System Restore (though it uses the same technology in the OS to achieve its goal), you can use this to roll the installed driver back to the previous version.

What can also be useful is the *Driver Details* button; see Figure 9-7. In Chapters 14 and 15, I will detail how you get detailed error reporting information from a Windows PC. Sometimes, and especially in the case of something like a Blue Screen of Death, this will point to a specific file as causing a problem. If you want or need to check what files on the PC constitute the driver, this is where you can get the information.

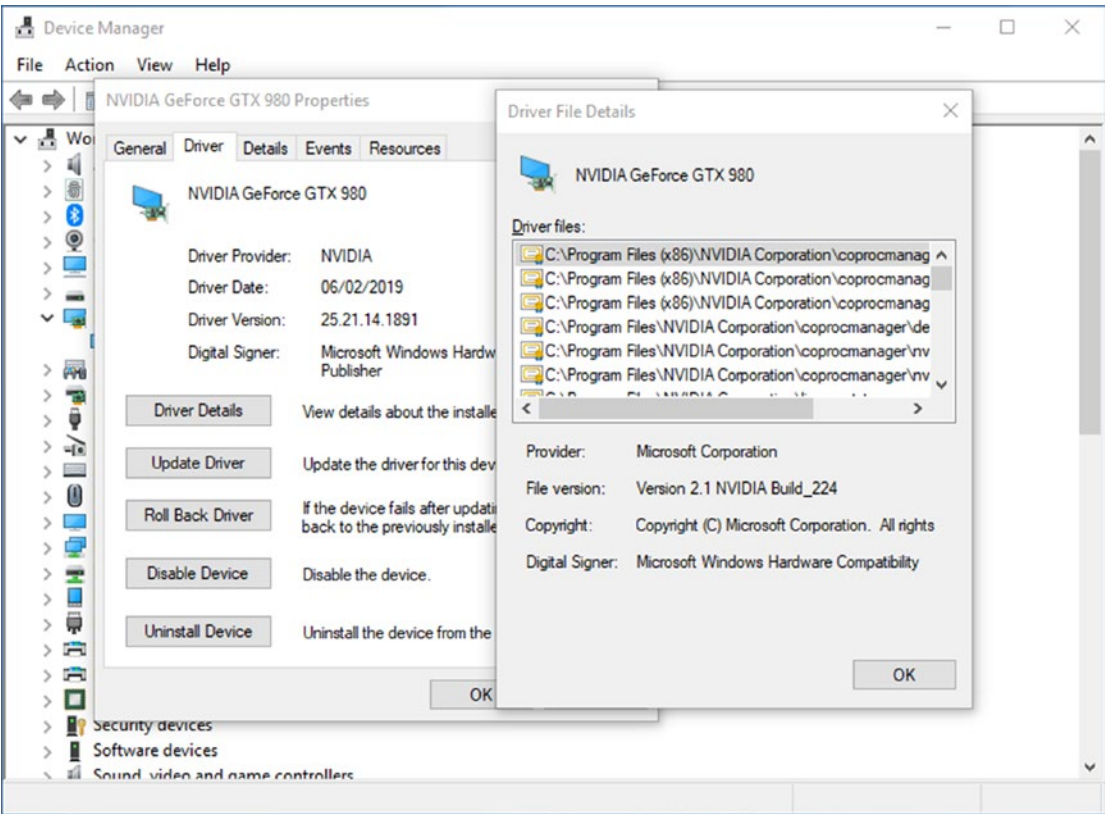


Figure 9-7. You can see what files on the PC constitute the driver

If you do need detailed information about a driver, which will most commonly be something like the driver version or release date, this can be found in the *Details* tab, where a drop-down menu is available that contains a huge amount of information about the device driver and its settings; see Figure 9-8.

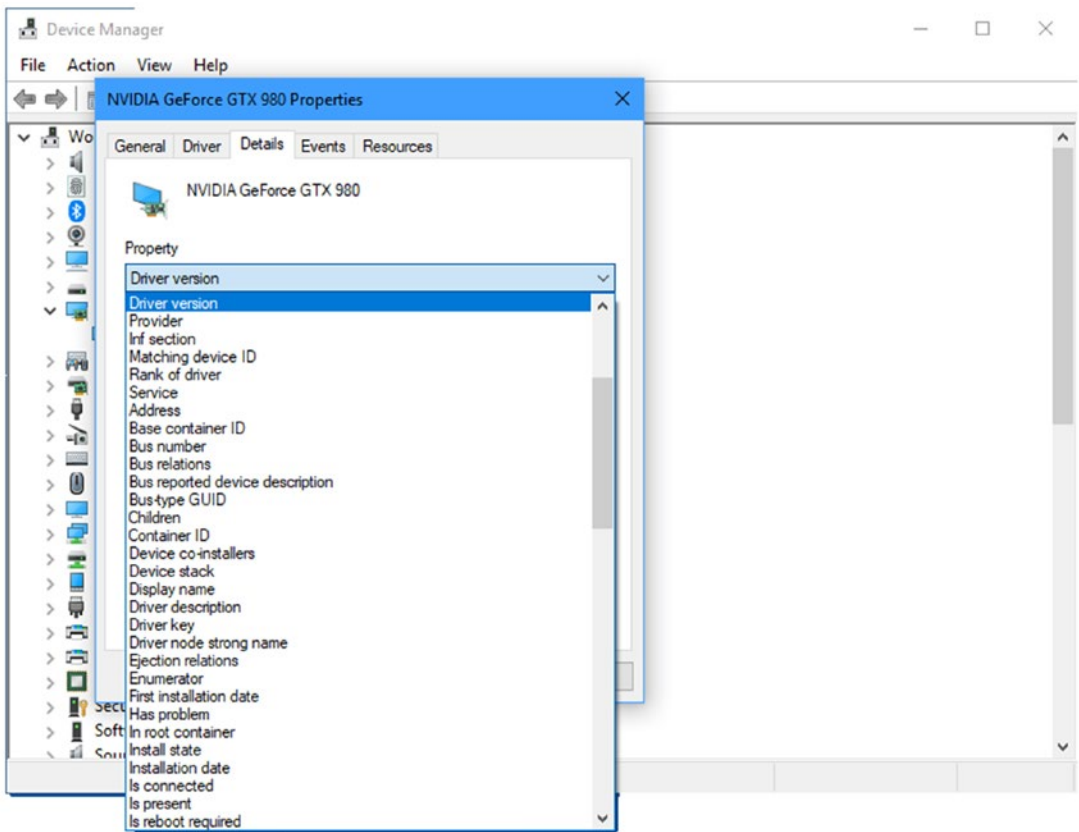


Figure 9-8. You can also obtain detailed information about drivers

If you need to see specific events associated with the device, then you don't need to hunt through the Event Viewer to find them. The *Events* tab in the device's properties details any events for which there is stored information; see Figure 9-9.

The information available here isn't anywhere near as detailed as that found in the Event Viewer, and I'll show you how to obtain this advanced error information in Chapters 14 and 15, but it can be a good place to start if you believe a specific device, or its associated driver.

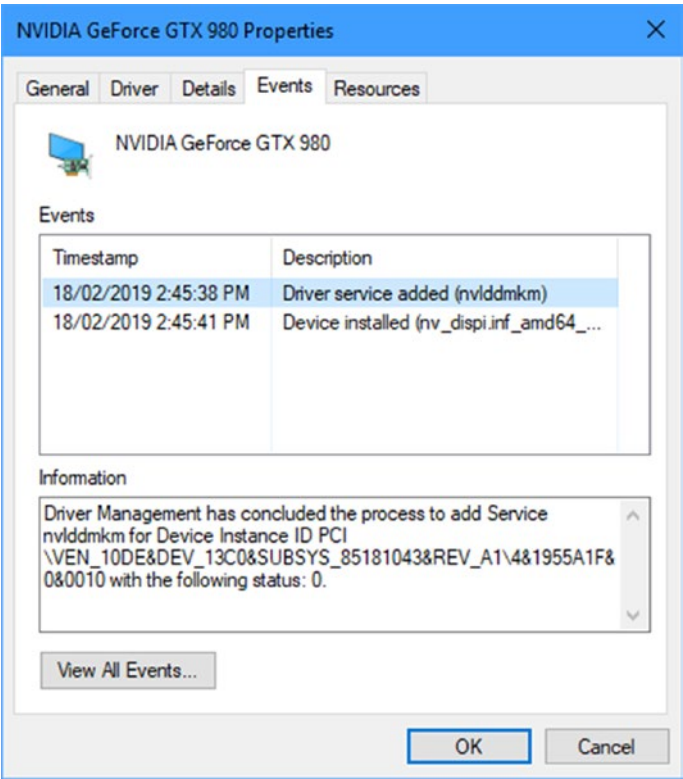


Figure 9-9. You can also see errors and events associated with the driver

What Else Goes Wrong with Peripherals?

When it comes to other problems that you can face with peripherals, these are commonly associated with the port the device is plugged into, the cable it’s attached with, or any plug attached to it or the cable.

For example, it’s especially common with printers and network cables for people to trip over the cable and pull and damage either it, the plugs on it, or both.

One useful little trick, though Microsoft have never publicly acknowledged this is an issue, is to unplug a USB device and then plug it in again but into a different socket. It’s very common for Windows to not see devices such as printers that appeared perfectly visible before. Unplugging them and reinserting the plug is not always the solution to the problem. Plugging the USB plug into a different socket however (sometimes you need to try two or three to find one on a different connection bus on the PC’s motherboard) can force Windows to reload the driver, making the device visible once more.

Summary

Peripherals can be a problem, especially those connected via cables or that are older legacy devices. While it's easy with USB devices to have them automatically configured when attached to a PC, setting up serial and parallel hardware can often appear to be a chore, especially as there's no way in the system to then make a backup copy of the settings.

Snagged cables and broken plugs though, while often the most visible way to diagnose a problem with a PC or device, pale into insignificance when compared to the issues you can face with the environment you find yourself in, and even the building you're using the hardware inside. In the next chapter, we'll tackle this prickly issue.

CHAPTER 10

Building and Environmental Factors

We tend to think that it's just IT equipment and users that cause problems with our systems, connections, and software, but that isn't always the case. In fact, there are a great many environmental factors, both natural and manmade that can be the root cause of the problems we face.

These are often missed simply because nobody expects an IT problem to be caused by the world around them, but it's worth examining what types of problems our environment can cause. We might not be able to do much about them, but there are always ways to mitigate against them.

The World We Live In

Earth is a beautiful planet. In fact it's often argued in science fiction movies that Earth has just about every habitat known to exist, from deserts to forests, oceans and snowy mountains, and more besides.

Normally that world doesn't affect our IT systems at all, because we use them in manmade buildings, with air conditioning, safe and protected environments, and stable electrical systems.

But what are the environmental factors, and the different environments on our planet that can cause disturbances, and how can we mitigate against them?

Weather

The weather is always with us, and wherever we go; there it is. Now you may be lucky enough to live and work somewhere with stable and predictable weather, perhaps even

where it's sunny most of the time. Most of us though live in places where fierce electrical storms can cause disruption, heavy rain can batter streets and buildings, and even a few inches of snow can cause major disruption.

It's always worth considering then where in the world the user(s) are, and what the weather may be at their location. To give an example, an electrical storm can fry just about any electronic system if it's not properly protected. You might think that this means that plugging electronic equipment into a surge protector will do the trick, and prevent electrical burnout, but this isn't always the case.

I, for example, am in the process of moving from the United Kingdom to the south of France as I write this (it's quite busy here). The United Kingdom has a great, very stable, and very reliable electricity single-phase grid. Power outages in the United Kingdom are rare, and the system is very robust, meaning that electrical surges are easily protected by a domestic surge protector.

Much of France though, especially rural France where I live, still uses a much older three-phase system. This system is much less robust, significantly prone to outages, and when surges occur, which can be caused by electrical storms in France as much as anything else, a surge protector is rarely enough to protect your equipment, and I've heard many stories from people who say they've had very expensive computer equipment destroyed even after plugging them into a surge protector.

The answer to these situations is twofold. Firstly, you can advise people to unplug all electrical equipment in areas with poor electricity networks in the event of a storm. This also includes unplugging cables such as phone/DSL and TV and satellite system wires.

Of course this doesn't work if a storm appears suddenly in the middle of the night, or when the person is away from the premises. For these circumstances the advice is to use an Uninterruptable Power Supply (UPS) with a built-in surge protector. Both the surge protector and battery work in conjunction with one another in this instance, with the battery stabilizing the electrical supply to help avert blown circuitry.

One thing to bear in mind if you are asking people to unplug cables when a storm is due is to be aware that the next time they call you for technical support, it might be that they've forgotten to plug all the cables back in again.

Sand, Dust, Water, and Moisture

There was a time when no sane person would take their phone to the beach, or leave it laying around on the sand. Sand is a killer for several reasons. First it is abrasive and can cause physical damage to electronics if it finds its way into cracks and sockets. Second it's an insulator, meaning it's not electrically conductive. This can result in connections being lost between components where even a single grain of sand interrupts the connection.

Then however there's wet sand. Wet sand, even with a tiny amount of water, is highly conductive, which can short out components if a grain comes into contact with them. I think you'll agree then that sand, in whatever form, can be a huge problem for users of IT equipment.

The other issue with sand is user ignorance. Most smartphones these days, and some tablets, are Ingress Protection (IP) rated. You will probably have heard of phones that are IP68 rated, for example. The two numbers represent the resistance of the equipment to solids and liquids, and they're measured on numeric scale.

This means that an IP68-rated phone has a solids rating of 6 and a liquids rating of 8. You can see full details of the solids and liquids ratings in Tables [10-1](#) and [10-2](#).

Table 10-1. *Ingress Protection (Solids) Ratings*

Level	Object size protected against	Effective against
0	Not protected	No protection
1	>50mm	Any large body
2	>12.5mm	Fingers or similar objects
3	>2.5mm	Tools, thick wires
4	>1mm	Most wires
5	Dust protected	Ingress of dust is not entirely protected
6	Dust tight	Complete protection from dust

Table 10-2. *Ingress Protection (Liquids) Ratings*

Level	Object size protected against	Effective against
0	Not protected	No protection
1	Dripping water	Vertically dripping water
2	Dripping water when tilted up to 15°	Dripping water at an angle no greater than 15°
3	Spraying water	Water falling at spray
4	Splashing water	Water splashing against the enclosure
5	Water jets	Water projected by a nozzle up to 6.3mm
6	Powerful water jets	Water projected from a nozzle up to 12.5mm
7	Immersion up to 1m	Submersion up to 1m for up to 30 minutes
8	Immersion beyond 1m	Submersion between 1m and 5m for up to 30 minutes

One thing to note with IP ratings, and this would usually only affect specialist maritime equipment but could also affect consumer and business equipment if used inappropriately at sea, is that water pressure begins to greatly increase below 5m, and modern electronics are not protected against this. If any equipment is to be used underwater below 5m, it will need a specialist protective casing, or need to be specifically designed for this purpose. This might be something to remember the next time a client goes snorkelling with his smartphone.

It's far more common for people to become exposed to things like dust and water than you might expect. For example, anybody who has to visit, or work on a construction site, or undertake a business trip to the middle east can face challenges using devices away from dusty or sandy environments, as it can hang in the air and be moved great distances sometimes by air currents.

Similarly, anybody working in or visiting a region of the world such as the Far East might face very high degrees of humidity, or even typhoon rain, both have a habit of penetrating just about anything, as anybody who's ever tried to walk a short distance in this condition will attest.

It may be that you request specialist equipment for people working in these environments, or traveling to them, such as a hardened, protected smartphone or a toughened laptop.

People need to work, and as such they will take risks assuming that “everything’s going to be fine” when it really isn’t. This means that laptops that are really unsuitable for these environments will be used there anyway. This is as much a staff training and awareness issue as it is anything else and should be factored into any training and information packs that you offer.

The Built Environment

Anybody who has tried to use a satnav in a city center, surrounded by skyscrapers, or that has tried to use a cellphone at a music festival will attest that it’s not always easy, or even possible, to get a signal. This can be extremely annoying, but it’s by far the end of the problem.

Our modern societies and the built environments in which we live can present real challenges for our use of IT equipment, especially when people move or travel from one place to another, and find that something that worked perfectly well yesterday suddenly doesn’t work any more.

Wi-Fi, Where-Fi Art Thou?

Were you using tech in the early days of Wi-Fi? If you were, then you’ll remember that it was pretty awful. The range was miniscule, the coverage was worse, and the stability of the connection was unreliable on a good day.

Things have improved considerably in recent years, and a strong 802.11ac connection now can easily reach several hundred feet to the bottom of the garden. In open spaces though there aren’t problems with Wi-Fi connections, but there can be problems caused, and it’s especially bad for people from the United States who come to Europe; let me explain.

In different parts of the world, the construction standards and materials differ greatly. For example, in the Far East, the building standards are much less rigorous than those in the West. What’s more significant though are the materials from which builds are constructed.

In the United States, it’s very common for buildings, especially houses, to be constructed from wood. Beams and struts are made of wood, walls are made of wood, and as a result, housing can be cheap. It’s just not designed to last forever, and it can be common for a building to be demolished after only 20 years of use and completely rebuilt.

In Europe however things couldn't be more different. You simply won't find main buildings or dwellings constructed in this manner; they don't exist. Brick, breeze block, and concrete are the materials of choice and have been ever since the Roman Empire invented concrete somewhere around 25BC.¹ To make matters worse, many buildings in commercial use in Europe still are made of thick stone, sometimes more than 1 foot thick.

Nothing blocks a Wi-Fi signal like thick stone and brick walls. If you have a US-based worker on a trip to Europe calling to complain that they can't access Office 365, it could very well be that they're in an environment where the Wi-Fi signal is severely restricted because of the materials used in the building they're sat in, and the method of construction used.

Bluetooth and Cellular

The problems facing Wi-Fi signals also affect Bluetooth and Cellular signals. When Bluetooth first began to appear, it had a range of barely 1 meter, and as such it took several years and several upgrades to the technology before widespread adoption took place.

These days, Bluetooth signals will easily reach 10 meters, and some will even stretch up to 100 meters, and the adoption of Bluetooth devices has exploded as a result. People finding that their devices suddenly don't work however can often be the result of the built environment, and the materials used in construction. This can be especially pronounced if they move around in the building while using Bluetooth.

With Cellular there are the same problems, but most people are sensible enough to recognize when they have a poor signal and to relocate to a place where the signal quality improves. The challenge with Cellular is one that is yet to come, as when we have fully 5G adoption (which may have happened by the time you read this) people may start to experience signal degradation and dropout that they don't expect.

¹The earliest recorded use of concrete dates back to around 25BC where the Romans, annoyed that their great halls needed to have supporting pillars every few metres, sought a substitute. New halls and palaces constructed using this new material enabled architects and builders to create wide open spaces for meetings and the enjoyment of those who visited. This concrete was of such a high quality that many of these buildings are still standing and in use today.

This is caused by the frequencies used for 5G radio signals only supporting short distances and being pretty poor at passing through anything, even a closed window. The technology companies providing 5G equipment are fully aware of these problems and are working to enhance the signal and range of the technology, so it will improve over time.

Conversely, other companies are binding 5G signals with existing 4G networks so as to pick up the slack, so a Cellular connection will automatically switch to 4G when the 5G signal drops out.

This isn't the type of information you might expect "normal people" to be aware of however. They will see 5G as being a substantial upgrade on 4G (LTE), and as a result it's not unreasonable to expect coverage to be better, streaming to be faster, and so on. It's just one more thing to be aware of with the people you support, especially given that there will very likely be different variations of 5G used in different parts of the world.

Cities and the Countryside

How many people do you know who live in cities and almost never visit the countryside, or people who live in the countryside and almost never visit cities? These types of people are much more common than we might expect, and they can throw up their own support challenges.

The problem people have with IT these days is that they just expect it to work. They think that it'll work in the same way their TV does, or their microwave oven. This has been the case for several decades. There was a time, back in the days of Windows 95 and Windows 98, when it most certainly was not true. You might have been reimaging a desktop PC almost every week in these dark times, due to the unreliability of desktop operating systems.

Now things are very different; it might even be possible for a computer to go its entire life without a single reinstall taking place (I said "might"). People will still view their computers, laptops, smartphones, and ultrabooks though in the same way they view consumer electronic devices in their home, like the TV. You will know as well as I do that they're not.

In the exactly the same way as people just expect their tech to work, they also expect things to "just work" wherever they happen to take that tech. This isn't always the case however and tech that might work perfectly in one location, might encounter problems when transported to another.

Previously I've mentioned the problems people can face using satellite navigation in cities. This is because satnav is a line of sight technology, and the tall buildings and skyscrapers found in the center of cities can simply block the signal.

The same thing often happens in the countryside with satellite connections that are blocked by tall trees and woodland. If there's no line of sight connection, there's no connection.

This is why when people contact you for IT support, it's vitally important to ask the question "what's changed?" The person telling you that something worked perfectly well yesterday and has always worked perfectly but now suddenly doesn't work any more can take on an entirely new dimension when you discover that they've left their coastal office in Maine and traveled to Rome on business.

Summary

The environment in which we live can have a dramatic effect on our hardware, how it works, whether it will work at all, or whether it can be destroyed by leaving it on a beach, or even plugged into an electrical socket at night.

This brings to a close our discussion about the problems we face with technology, and what can cause it to cease working. I did mention just a little while ago the importance of asking the right questions of people who contact you. In the next chapters, we'll look in-depth at this questioning and the paperwork and procedures involved in providing top-quality IT support.

PART IV

Documentation and Reporting

CHAPTER 11

Why Good Documentation Matters

I'm an author of technology and computer books and manuals, so you might expect me to say that good documentation in IT matters and is important. It's not just about my personal opinion however, or that of my publisher, other tech authors, technical writers, and editors. I'd suggest that it's vital, if I may be so bold, to have good-quality documentation for your IT systems for everything from staff training to specialist training, and especially for everybody involved in the IT troubleshooting and repair process. Let me explain why.

Documentation Saves Time and Money

The most important factors to any business are time and money. People are mostly paid by the hour, and for those hours the business has an expectation of how much work will be accomplished in that time. Some workers will have set workload expectation they must meet, for others things are a little more flexible but their performance and productivity will still be monitored over a period of time.

It might be easy to justify this, or explain it away on the grounds that businesses don't want slackers working for them. This is a conceit, as we all know or have met people working for businesses who do exactly that. It becomes important then because the business will have costs that will be allocated in order to achieve a certain level of turnover, and these will be carefully calculated. In some businesses, especially high-pressure industries, the turnover expected from each employee will be high, perhaps because the margins are small. In other businesses the amount required from each employee is smaller, possibly because the actual financial value offered by each employee is much greater due to their expertise or training they have received.

Whatever industry you work in, or even if you do work for yourself or study at college, time is valuable. You don't need to spend a lot of time figuring out how to do A or B. This is where the phrase RTFM came from, which for the uninitiated stands for Read the [Expletive Deleted] Manual. People don't read manuals because they either don't have the time, or don't think that it would be worth their time to do so, and this is why almost every new product now comes with a one-page quick-start guide instead, with the manual downloadable from the manufacturer's web site. If you thought this was all about saving money, then you might be surprised to hear that it's not really about that at all.

What you need is for people to read the manual, or at least identify the value in doing so. I wrote the manual for the Gemini PDA handheld computer, **pcs.tv/2X8RTE4**; I knew this, but I also knew that the developers of the Gemini, UK-based Planet Computers, wanted the manual to be available both as a free downloadable PDF and also to purchase in paperback for those that wanted it in that format.



Figure 11-1. *The Gemini PDA*

Now the Gemini is a special case; it wasn't a mass production, mass appeal device; and its enthusiast audience would therefore be much more likely to read the manual in full. In order to ensure a quality experience for them, I made sure to include extra information that would “add value” to that experience for them.

This meant including information during the chapter on setting the device up, and creating a Google account, on how to create secure passwords and how to use two-factor authentication. Then throughout the manual, while writing about the features available both with the hardware of the device and the operating system and apps that are installed, not just what they're for and how to use them, but how the reader could best leverage those features to become more productive and happier with their overall experience.

Then there's making sure to include appropriate information on what can go wrong, how you can repair common issues, and what you definitely must not ever, ever do; which in the case of the Gemini includes things like removing the battery, or taking the keys off without being extremely careful.

Documentation for Training

So documentation is important. It saves you time and can drastically reduce your workload, and it helps other people. For all of this to happen though, it has to be written correctly, be in the right format, be the right length, and be appropriate for the audience.

This is especially important when it comes to training materials, as you'll either have just a very short time to impart vital information to people, or you'll need to provide something they can work through in their own time (or both).

There are things you can consider using and doing when writing training materials that can often really help the people you need to read them, and these include the following:

- Using larger text with bigger gaps between text rows. This can be of great help in digesting information for anybody but is of a particular help to older readers.
- Using numbered and bulleted lists can help people learn in a step-by-step manner that makes easy individual step easier to understand. Don't put too many actions in a step though, keep it to just one or two; otherwise what you're doing will be counterproductive.
- Using screenshots and images, especially when making step-by-step instructions, can really help people visualize what it is you are talking about.

- Putting a few questions about the training at the end of the document, or at the end of each section of the document, will help the reader consolidate their learning. You need to make sure they have access to the answers immediately afterward, but they will immediately be able to see if they have missed something or made a mistake, so they can return to that part of the document and review it.
- If you ask for feedback on the documentation and training, don't ever ask in the format "one to five stars" as you're very unlikely to get any meaningful data. What happens in this case is that most people will either just put five stars because they want to be nice, and show appreciation, or they'll put three stars which is right in the middle, as they really don't want to have to think about it. You should ask for a scale of one to ten, or one to eight, as it forces people to think more about it, and then perhaps have a couple of questions such as "What do you like/find easy" and "What didn't you like/find difficult."

Dumbing Things Down

Several times throughout this book I've said that you can never be sure who it is that you'll be providing support and help to. This goes doubly for training materials as, while they'll be distributed to staff in "X" department as a generalization, unless they're in a training room with you, you won't have the first idea who they are, where they are in the world, what language(s) they speak, what their level of education or technical understanding is, and so on.

You might hate it when the TV or online news is accused of "dumbing things down" for the audience, but it's actually required far more often than you might believe. In the case of the media, they have viewing figures or readership levels to maintain. They will know and have done research into who their audiences are, and they'll have some (I stress only SOME) idea of who these people are.

As an example, probably a bad one but I'll use it anyway, the people who read *The Washington Post* will generally speaking have a higher educational level than those who instead watch FOX News on cable TV, or who read their local newspaper. It could also be said that people who read and subscribe to an industry journal will be more technically minded toward the language of that industry than someone who gets that industrial news from documentaries on the National Geographic channel or a Sunday newspaper magazine.

Because the people you are writing training materials for could be *any* of these, you need to make sure that the language you use is as accessible to people as possible, and you can do this in different ways depending on your own needs and your own opinion of which you think is the right way to jump.

- Choosing your language carefully is often a good way to go. This can be done by avoiding technical language and jargon as much as possible, or by explaining jargon and acronyms by detailing the full meaning of that acronym in brackets immediately afterward. This is something both myself and Apress do in our books.
- Adding a terminology list to the end of the training materials that contains the full names and plain text descriptions of technical terms can be a useful reference.
- Similarly, you could add appendices of useful, but additional to the training, information to the end of the training materials, perhaps including a list of further reading for those who want to use your training as a stepping stone to further learning.
- Trying to avoid language that uses more than three syllables can be useful but is not always suitable. Firstly because it can often be very difficult not to include a great many four syllable words but also because this really does dumb things down. If you believe you will be providing training however for people who speak your language as their second, or perhaps even their third language, this approach can improve their understanding of the training materials.

Documentation for Troubleshooting

Once you have trained people, you still have to assume that things can go wrong. In the next two chapters, we'll look at how you actually structure good diagnostic and troubleshooting paperwork, but first it's important to look at why paperwork for diagnosis and reporting needs to be structured in a certain way.

Just as with creating training materials, your diagnosis and reporting paperwork needs to be suitable for the audience who will read and have access to it. If you are the only person who provides support, perhaps the only one in your organization, then

you'll likely still need paperwork for recording and auditing purposes, but things can be worded in ways best suited to yourself. If you work as part of a team, then there are considerations to make.

Personnel and SLAs

The first consideration is your own availability to work on a troubleshooting project. We all work long hours, and work regularly (I know, it's awful), but think back to the last time you took a vacation, or a day off because childcare wasn't available, or perhaps some time off because you were sick.

Any business providing IT support services will have in place at least one SLA (Service Level Agreement). This will stipulate how quickly a job will be picked up and responded to, and under what timescale it will be resolved. These SLAs are crucial to securing support contracts as any businesses or organization seeking support will need to know that their problems will be resolved quickly and efficiently.

It's one thing therefore to take on a support job on a Monday and resolve it by Wednesday afternoon, but quite another to duck out for a doctor's appointment on Tuesday for an hour, to be told you're going to have to take the next few days off work. Alternately, you might find that you take on a support contract on a Friday, but have far too much fun with your friends when you're all out on Sunday, and then for you to need a duvet day at the start of the next week.

We need to be realistic about the support we offer, and about the time we can guarantee we have available to support those cases. It's for this reason that we need to structure paperwork in such a way as anybody that you work with can pick it up and very quickly bring themselves up to speed on what the problem is, what's been done, what's needing to be done, and what solutions you might have put in place already. Quite simply, the alternative to this is that whoever takes on the case has to begin again right from the start, which is of no help to anybody and will only annoy the customer.

Getting in Line

Another reason to have well-structured paperwork is because you will probably not have been the first person to handle the case, and likely won't be the last either. It's very common for a first-line worker to take the details from the end user and walk them through some diagnostics and basic fixes for common problems.

Alternately, you might be a first-line worker who can't resolve the case yourself and needs to pass it on to somebody with more experience. This is not a problem as we start in first-line work, and you may see issues with the current way the paperwork is structured or believe that adding or removing some items from it can increase efficiency for everybody. Any good company that provides support will encourage and value feedback from every worker.

Engineering Solutions

Many problems can be resolved either by the first-line worker talking the user through some basic steps and fixes or by the second-line technician taking remote control of the problem computer or by giving the end user some more detailed instructions to follow.

Sometimes though you just have to send an engineer to sort out the problem. If you're on site already and you visit the user themselves, then you can diagnose, troubleshoot, and repair all in one visit (hopefully). Engineers though face other challenges.

These people will have a workload that's dictated by several different factors. Firstly, you will very likely have a logistics team responsible for assigning engineers to jobs. They'll make sure that each engineer has jobs to go to that reflect a single geographic area, and the travel times between sites, the complexity of the job, and how long you estimate it will take to repair that job. These people will also order and deliver to the engineer any new hardware or other parts that you have specified will be required to fix the problem.

All of this can very quickly get snarled up by the wrong part being delivered, or no part being delivered, the engineer getting stuck in heavy traffic (which can happen to anybody at any time).

If somebody then makes an incorrect diagnosis, perhaps because they weren't given valuable information, the engineer has to figure out the problem for themselves which will slow them down further. Remember that the engineer will very likely be making calls that have been dealt with by your colleagues, as well as yourself.

With all this to contend with, that engineer needs to be able to get up to speed with the problem very quickly when they arrive on site. This means that paperwork absolutely must be clear and concise, with easily understandable instructions of what they're to do, and clear details of what's been checked and done so far, so they can act appropriately if the pre-determined solution doesn't work.

Keep It Clear and Concise

You will see then that it's crucial for a whole host of reasons that paperwork should always be clear and concise. It shouldn't be cluttered, it should follow a very clear and logical structure, and it most certainly should not encourage support technicians to write long and rambling monologues about what they think is happening and why, as nobody has time for any of that.

This gives your paperwork a very clear beginning (first-line support), middle (second-line support), and end (engineer and conclusions). It's true that this paperwork will very much follow the narrative of a story as well. In a good story, it's always the case that you have a reasonably short introduction to the protagonist, the antagonist, other major and minor players, the setting, the lead up to where the story begins, and so on.

The end of the story will draw everything that's happened into a series of events or conclusions that allow the protagonist to figure out what's happened and take the action necessary to rectify the problems they have faced thus far.

It's the middle of the story that contains all of the meat. It's here that multiple scenarios will play out, the characters will interact, events will occur, and problems will arise that the protagonist and the other characters in the story will need to deal with or avoid.

This is exactly the same with IT support paperwork. The beginning is the diagnostics conducted by the first-line support department, with their conclusions. The end is the engineer report on that they had to do, to resolve the problem, or the report on what second- or third-line support did to resolve the problem if an engineer wasn't required.

The middle is everything else. The detailed information sought from the end user, the diagnostic steps that are undertaken, and fixes that have been attempted. It's the meat of any good story, and the information required to make sure that appropriate fixes are implemented and that changes to procedures, policies, training, or software and hardware are made.

Summary

In the next two chapters, we'll look at these beginnings, middles, and ends in detail, and I'll give you examples of how you can structure and layout paperwork so it is suitable for your own requirements and for the hardware and software you support.

This will include creating good troubleshooting guides that people can follow to diagnose problems with hardware and software quickly and efficiently, and then creating more full-featured paperwork from which anybody can track the progress of an issue at any time or from which any personal unfamiliar with the case can take it on board and quickly bring themselves up to speed.

CHAPTER 12

Creating Troubleshooting Guides

How do you go about creating paperwork that can help you manage your workflow, both individually and across teams providing IT support? Throughout this book I've detailed the importance of being methodical, defining a clear set of procedures that enable you to achieve your stated aims and goals without fuss, and documenting everything along the way.

This last point is incredibly important when you create troubleshooting paperwork. This is because there is every chance that you won't be the only person reading and dealing with the paperwork, or the client. Have a think at the different people in your company or organization that might be working directly with your client, and some of the circumstances in which this would happen. These fall broadly into two different categories.

- **Structured teams** – where you will have dedicated personnel to manage first-, second-, and third-line support, engineers, and logistics and accounts/parts personnel
- **Days off, vacations, and sickness** – where you will need cover for your own role for any days you have away from the business and any vacations or time off you need off for sickness

It's also worth noting that in these two structures, there are things that can *never* be predicted. Taking sickness as a good example, you don't know when you're going to be struck with the flu, or when your youngest child might suddenly contract chickenpox or fall out of a tree they were climbing. It does go further than that however as there's also no way to know when you'll be snowed in, have the car break down on the highway or, frankly, just need a duvet day for whatever reason.

It's no simpler within teams as the person, or one of the people who has been working on this case with you might also need a duvet day, or have a child that got slimed by a frog (I'm not actually sure the latter counts as a good reason for a day off work, but it sounds suitably icky!). You also don't know when a member of staff will leave, or a new member of staff will join.

This last example makes it doubly important that your paperwork is easy to follow and easy to use. A new staff member, no matter how experienced they might or might not be, needs to be able to pick up and run with your paperwork from the very outset. Time is money as the old adage goes, and having to spend excess time being trained on the paperwork cuts into the time you have available to fulfill your SLAs (Service Level Agreement[s]) with the customer.

Clean, Concise, and Easy to Understand

When you're creating your paperwork and troubleshooting guides, then you should structure them as though they are a flowchart. You *could* create an actual flowchart, but this isn't very useful unless you are programming it into a computerized questionnaire, where people are guided through various different threads depending on the answers they provide.

It's actually worth noting that I don't recommend computerized question and recording systems per se. Sure, they're efficient, easy to use, and certainly more environmentally friendly than paper, but they also have several distinct disadvantages.

Chief among these disadvantages is that nobody is actually able to glance at them and get a quick overview of the situation, what's been asked and done, what's required, and what people have recorded. It's a sad truth about computer screens that they're just not as easy to use as a few pieces of paper. You can't lose the information quite so easily, or spill coffee on it (though many people do try their best), but there are large volumes of scrolling, clicking through different pages, options, menus, and buttons, and all of this can very often leave you forgetting what it was you read back at the beginning.

If you *do* decide to use a completely computerized system, it's well worth making it look like paperwork, both because it'll be simpler for people to understand and because it makes it easier to print in a format that people will want to easily digest and carry around with them.

Printing is often overlooked when designing bespoke IT recording systems and software, but it's well worth spending time planning and considering. You might be surprised at just *how* useful that printout, even an occasional one, might be.

Flow Logic

So how do we go about designing paperwork that looks, feels, and works like a flowchart? The easiest way is to separate the different types of information, or different processes, into groups. In Figure 12-1 you can see one example for first-line support paperwork, where everything is clearly separated, easy to follow, and simple and straightforward to complete.

1st Line Support Checklist

Problem area

PC OS	<input type="checkbox"/>	Printer	<input type="checkbox"/>	Other (Specify)
PC Desktop / Update	<input type="checkbox"/>	Other Peripheral	<input type="checkbox"/>	<input type="text"/>
PC Software	<input type="checkbox"/>	Networking / Internet	<input type="checkbox"/>	
PC Hardware	<input type="checkbox"/>	Cloud Service	<input type="checkbox"/>	

When did the problem occur?

More than a week ago	<input type="checkbox"/>	Today	<input type="checkbox"/>	Recurrence	<input type="checkbox"/>
A few days	<input type="checkbox"/>	Just now	<input type="checkbox"/>		

Has the user tried any of the following?

Restarting the PC / Device	<input type="checkbox"/>	Replicating the problem on another PC	<input type="checkbox"/>
Checking cables / Power	<input type="checkbox"/>	Seeing if anybody else has the same problem	<input type="checkbox"/>

What's changed recently?

The OS / Software was updated	<input type="checkbox"/>	Something new has been installed	<input type="checkbox"/>
The user has changed something	<input type="checkbox"/>	Nothing	<input type="checkbox"/>

Figure 12-1. *It's crucial to make paperwork easy to understand*

This paperwork is separated into the main question areas ... What is the problem, when did the problem occur, what has the user already tried, and what is it (if anything) that has changed recently. We’ve covered these question areas at length in this book and so I won’t detail them again, but just this small amount of basic information, recorded and formatted appropriately, can give just about anyone a huge head-start on the troubleshooting process.

Next up is the main troubleshooting section. Now, this will vary considerably depending on what type of hardware/software/services you are supporting. In Figure 12-2 you can see one example, in which I have separated hardware, software, and other types of computer issue.

Quick Tests / Questions!

Hardware

Restart PC / Device ☐

Electricity supply ☐

Error / Power light ☐

Cables ☐

Sockets ☐

Battery ☐

Installed hardware ☐

Missing hardware! ☐

Recently moved ☐

Other (e.g. UPS) ☐

Age of Device ☐

Cleanliness ☐

Software

User install ☐

Recent install ☐

Recently updated ☐

Experience of use ☐

Any other PCs ☐

Network access ☐

Internet access ☐

Installed on PC ☐

Local / Domain Acc ☐

RemoteApp ☐

Legacy app ☐

Peripheral problem ☐

Other

Any other PCs ☐

Location of User ☐

Location Change ☐

Cloud Services ☐

Log out / in ☐

Reported Issues ☐

Maintenance ☐

Encountered Before ☐

What's changed? ☐

Knock-on Effects ☐

Local Reset ☐

Network Reset ☐

Figure 12-2. Easy to use checklists are always helpful

Each one of these follows a simple flow logic design for things that can often help fix a problem, and in the order you might want to try them. Naturally with hardware this begins with “have you tried turning it off and on again?” the age-old chestnut IT support question, moving on to checking the electricity supply, lights on the unit, cables, sockets, battery if it has one, installed and perhaps missing hardware, and so on.

With software we begin with “is this official company software, installed by the company?” as many users will call support because the piece of software they installed themselves, and which they weren’t supposed to, isn’t working.

The Dev Problem

It’s worth noting here though that software, app, and web developers can cause some headaches for support helplines, as they are frequently allowed to install their own *logiciel de preference*, or software of choice. This can also sometimes extend to graphic artists, video and photo editors, and engineers.

Because some programmers (commonly known as devs) like to program using either just a command line, just a wyswig (what you see is what you get) interface, or a mixture of both, and they often feel more at home with a completely different software setup than the person who sits next to them, you can sometimes find yourself supporting software which is not provided by the company, or for which you have no knowledge or experience.

This isn’t necessarily a problem as the main troubleshooting formula of taking a step-by-step approach still applies, and all the questions you would ask would be the same. It’s a lesson though in how you should be prepared for literally anything.

Now Let Me Tell You a Story ...

You’ll see in Figures [12-1](#) and [12-2](#) that there is a clear, beginning, middle, and end structure going on with the paperwork. It’s essentially a story that introduces our protagonist early on and details the challenges they face and what they’ve tried to do to overcome that problem. Your aim is to come into the story as the knight in shining armor who, rather than slaying the dragon, helps and guides the protagonist through the problem, giving them both a realization of the problem and a resolution at the end.

With this in mind, can you guess what the next section of paperwork would be about? We've dealt with what the problem is, and we've talked the user through some basic steps. Next up we want to record any additional information from trying those steps. It's perfectly fine to check a box to say something has been tried and didn't work. If it throws up some anomalous event or trigger though, then this should be recorded in a "What the hell just happened there?" box, or words to that effect.

The Story Continues ...

How you structure the paperwork from here will vary dependant on your own circumstances, as I've already said. It is very important however to stick with a consistent style and format, and certainly never to overcomplicate matters.

Think of this again as a story. How many times have you read a book or watched a movie where the plot was overcomplicated and far too difficult to follow? I can think of *Inception* (Warner Bros., 2010) as being one film I've been completely unable to follow no matter how hard I tried. It didn't mean the film was awful, as millions of people have enjoyed it enormously, but can someone who is completely unfamiliar with it just jump in and know what's going on? Probably not.

It's best then, if we're carrying on the movie analogy, to think of your paperwork as more of an Adam Sandler, or Jennifer Aniston film, where you can come into the film half way through but still quickly pick up that he's interested in her, she's got awkward parents, and she's worried that her job is going to take her out of the country just at a point when she's found love for the first time in years. These types of films might be dumbed down, but at least they're easy to pick up and digest.

What and Why

Jumping back then to what comes after the lists in Figure 12-2, you need to provide two types of information. What and why. The what answers the questions of what's been tried, what hasn't been tried, what got a good result, what made things worse, what might be required to implement a fix, and so on.

The why fleshes out each part of the what section with information explaining why something worked, why it didn't, why it was really a terrible idea, or why such and such ought to be tried in order to rectify the situation.

Should every check box and every question come with a “Why?” box in which you can respond with more detail? I wouldn’t recommend it as to do so would be completely counter-productive and result in paperwork that’s bogged down in irrelevant details, and that would be impossible to follow. Adding a few “Extra information” sections is always a good idea however, as allowing people to explain their reasoning for things or to flesh out detail can never be anything other than useful.

So Does the Princess Kiss the Frog?

We’ve already established in this chapter that being slimed by a frog probably won’t kill you, so it’s not unreasonable to ask if the princess (or the prince for that matter) gets to kiss the frog at the end of the story.

Every story needs a resolution because we all want to know how it ends, if it’s happy (which it hopefully will be) or if it ultimately is bittersweet. This extends to IT support where you need to know the resolution for three very important reasons.

- **Billing** – It’s much simpler to bill a customer for solving a problem, if you’re actually able to point to evidence that you solved it.
- **Tracking** – What would happen if the paperwork wasn’t marked as resolved and somebody in your workplace who has just been on maternity leave picked it up? They might think “This has been ongoing for ages, I’ll have to try and solve it.” This helps nobody and all you’re really doing is wasting everybody’s time.
- **Building a repository** – The data you get back from each case can be compiled into a list of common problems, common fixes, and can be used either to enhance your paperwork or processes in the future, or for staff training purposes.

This last reason is why there should always be somebody at the end of the support chain who can review each case, determine whether it should be just filed, or whether there’s something there that can help improve processes, build knowledge, or enhance workflow.

Having this additional information at the end of each job can very often reduce the need for the same processes to have to be completed again for the same or a similar problem for another user (or users). Once you have identified something that might be common, or that perhaps required an unusual fix, making staff aware of it through meetings or training can be invaluable in cutting costs, saving time, and becoming more efficient and effective in what you do.

Summary

It's very true that IT support follows a simple, "if this, then that" structure and that it's always good to reflect this in your paperwork. Another commonly used phrase "keep it simple, stupid" – otherwise known as KISS – applies to support. All of this means that the paperwork we create to help troubleshoot problems should do two important things. First, it should be simple and straightforward to follow. Second, it should provide space where additional, relevant, and pertinent information can be provided, without encouraging people to write endless reams of wordy notes that nobody wants to read.

With this in mind, we'll move on in the next chapter to how we create and manage paperwork for the whole process, from the first responder to the engineer and management, and examine how we can create consistency and detail, while at the same time making tracking and training as simple and straightforward as it can be.

CHAPTER 13

Creating and Managing Paperwork

In the previous chapter, I showed you how to create troubleshooting guides that can both help guide support personnel through common problems and their associated fixes while also maintaining a simple and easy to understand structure. This enables anybody to be able to pick up and run with a troubleshooting case, and it enables you to cover staff absence, vacations, and staff changes.

It's not just the initial troubleshooting that you need to consider the paperwork for however, as the whole case, from the first response to the final fix, needs to be treated in the same manner. There's just no way to know if you'll have the same person picking up and running with a case all the way through, as there are so many variables that can prevent this from happening.

In addition to the circumstances I mentioned earlier, you might find that a second-line support person who picks up the case is lacking the specific expertise they need to troubleshoot and diagnose the problem. This is quite common, and in these circumstances they'll often turn to a colleague who has better specialist knowledge.

That person then needs to be able to be brought up to speed on the case as quickly as possible. This means that *every* aspect of the case needs to be clear and concise for them, everything from the person reporting the problem and the hardware they're using to the step taken so far in diagnosis and to try and repair things.

First-Line Support Paperwork

In the last chapter, we looked at checklists and the types of processes you can ask your first-line support personnel to go through to quickly diagnose common problems or to obtain as much useful information about the problem that they can, as quickly as possible.

The rest of the paperwork, which is where the main reporting and detailing of the case goes, is every bit as important. Again this is something where it’s crucial to create paperwork that’s easy to use and easy to follow, though clearly the example given in Figure 13-1 will differ from your own, as you will support specific hardware and software and have specific needs.

Customer Name

Address

Phone Number

Problem area

PC Operating System

PC Desktop

PC Startup

PC Crashing

☐ Software

☐ Printer

☐ Other Peripheral

☐ PC Hardware

☐ Network

☐ Internet

☐ Cloud Service

☐ Domain

☐ Other (Specify)

Identifying the Device / Software / Service

Asset Tag / Serial No. / App / Service

When did the problem occur?

More than a week ago

A few days

☐ Today

☐ Just now

☐ Recurrence

Figure 13-1. *The most crucial information comes first*

Here we can see that we’re asking the most important questions first, the fundamentals of all IT support. Who is having the problem, what the problem relates to, and when did the problem occur or start?

Next is to move on to “what’s changed?”. In Figure 13-2 we can see that the form is structured around asking if any updates have been applied, what the user might themselves have done, or tried, and what else might have changed.

Has an update / maintenance been announced?			
Yes, for this product	<input type="checkbox"/>	Yes, unsure what for	<input type="checkbox"/>
Unsure	<input type="checkbox"/>	No	<input type="checkbox"/>
What's changed recently?			
The OS / Software was updated	<input type="checkbox"/>	New hardware has been installed	<input type="checkbox"/>
The user has changed something	<input type="checkbox"/>	Hardware has been swapped	<input type="checkbox"/>
New software has been installed	<input type="checkbox"/>	New location / PC has moved	<input type="checkbox"/>
Unsure	<input type="checkbox"/>	Nothing	<input type="checkbox"/>
Has the user tried any of the following?			
HARDWARE		SOFTWARE	
Restart the PC / Device	<input type="checkbox"/>	Restart the PC	<input type="checkbox"/>
Check for power / error light(s)	<input type="checkbox"/>	Log out / back in with correct ID	<input type="checkbox"/>
Check the electricity supply	<input type="checkbox"/>	Check other software works	<input type="checkbox"/>
Check cables are plugged in and not damaged	<input type="checkbox"/>	Check other PCs (if possible) for the same error	<input type="checkbox"/>
Check sockets are not damaged	<input type="checkbox"/>	Check for network / Internet / Domain access	<input type="checkbox"/>
Check battery is plugged in (if applicable)	<input type="checkbox"/>	Check for RemoteApp / Cloud app access	<input type="checkbox"/>
Check for loose components	<input type="checkbox"/>	Uninstall and reinstall app (if applicable)	<input type="checkbox"/>
Check for removed hardware / consumables	<input type="checkbox"/>	Reset the device (if applicable)	<input type="checkbox"/>

Figure 13-2. Checklists can be extremely useful

You might be surprised just how useful this form can be, as the simple process of moving a PC from one desk to another in an office can cause all manner of issues with routers and switches that are expecting it to be on a different internal IP address to the one it's now on.

Lastly, in this example anyway, comes Figure 13-3 in which additional highly valuable information is gathered, such as whether anybody else is experiencing the problem also, if the problem can be replicated on another PC (useful for remote support this one), and if either the customer can show support what's happening or if support is able to sign into the offending machine remotely.

Is anybody else experiencing the same problem?

Yes, reported to IT Support ☐ Yes, at customer location ☐ Unsure ☐ No ☐

Does the problem recur when replicated?

Plug the device into another PC ☐ Access the Service on another PC ☐ Unable to replicate problem ☐
Run the software on another PC ☐ Changing cables or socket ☐ Problem does no recur ☐

Do any of these apply?

Customer can provide screenshot / photograph ☐ Customer can reproduce via PSR-type app ☐
Support can log-in via Remote Desktop app ☐ Customer is able to follow diagnostic instructions ☐

More information is now required

Please detail the problem / issue

Figure 13-3. First-line support can obtain valuable information

You finally round this off with a box (clearly larger than the one shown in the image) where the first-line support person can add notes and make comments. You might wonder why there should be space for this, especially when that’s really the job of second-line support personnel.

The two reasons for adding space for the first-line support to add their own notes are that a) people who work in first-line support aren’t idiots (well, not universally anyway) and also that b) it’s very likely that the paperwork you create will cover absolutely every circumstance and eventuality.

Remember that your paperwork will be evolving all the time, especially as you support more clients, new hardware and software, and as problems arise that might be common but that you might not have predicted.

Second- and Third-Line Support Paperwork

The paperwork for second- and third-line support personnel will be broadly similar, though you would normally have a more free-form format for your upper-level support technicians, as problems go to them firstly because *they* are the ultimate experts but also because, let's face it, none of the processes you've worked through in the paperwork so far have resulted in the problem being solved.

You don't want to replicate anything that's already been covered, or information that's been provided by first-line support. While it's perfectly fine for security questions to be used to help confirm the identity of the end user, taking down all their details again is a waste of time, very unproductive, and will only irritate the person you're trying to help.

In Figure 13-4, we just go straight into troubleshooting, with common fixes that can be worked through quickly in some cases, or that the user might be able to try themselves, such as swapping a keyboard or signing in on a different PC.

Have any of these solved the problem?

User provides screenshot / photo	<input type="checkbox"/>	User swaps hardware	<input type="checkbox"/>	Other (Specify) <input type="text"/>
Talk user through diagnosis	<input type="checkbox"/>	User swaps cables / sockets	<input type="checkbox"/>	
Remote Desktop connection	<input type="checkbox"/>	User tries different PC	<input type="checkbox"/>	
User replicates via PSR tool	<input type="checkbox"/>	User asks colleague to replicate	<input type="checkbox"/>	

What has also been tried?

Step 1	<input type="text"/>	Step 2	<input type="text"/>
Step 3	<input type="text"/>	Step 4	<input type="text"/>
Step 5	<input type="text"/>	Step 6	<input type="text"/>
Step 7	<input type="text"/>	Step 8	<input type="text"/>

Figure 13-4. *Second-line jumps straight in with troubleshooting*

Here we start to get more free form. Remember that the people who work in second-line support are experts in their field and that if the steps you've detailed for first-line support had worked, you wouldn't have gotten to this stage.

Thus the next stage is to provide your support personnel with space where they can detail what else they have tried to solve the problem and what the results of each of those attempts were. You won't always have to detail in which order they occurred, but it's entirely likely that some problem areas will have steps that people will want to follow, in a logical flow.

It will be after the second-line and/or third-line support personnel have performed their tests and troubleshooting steps you *may* need to pass the case on to an engineer. This will be the case if either the fix has to be implemented on site, such as replacing some hardware, or if a solution to the problem has still not be found.

Engineers are extremely short on time. You need to always bear in mind that while anybody providing support from an office can switch quickly from one case to another (perhaps via the coffee machine), an engineer has to physically drive from one site to another. This takes a considerable amount of time, meaning every engineer's time is precious and the notes you provide to the engineers need to be a) concise, b) authoritative, and c) comprehensive.

In Figure 13-5 we see one such example. Again this follows on from all the other notes, and repeats nothing, but does allow the second- or third-line support person to point the engineer in the general direction they think the problem may lie, and details what (if anything) the engineer should be looking out for.

Notes for Engineer		
Diagnosis	<div> User has a problem with their printer, they say they have changed all the cables with one from a working PC across the office. This printer is not shared with any other user and the user isn't able to swap the printer with another on their PC to test as it's too heavy. No consumables are reporting EOL. </div>	
What to look for	Consumables, cabling, sockets	Hardware / Licences required
	Driver, software configuration	Good USB Type A > Type B Cable
	Hardware fault with printer	Printer Power Lead (HP M127fn)

Figure 13-5. Engineers are short on time, so they need clear notes

Additionally it's also worth pointing out that it's on this form that you need to inform the engineer what hardware, software licenses, or spare parts they should be taking. There's nothing that wastes time more than an engineer arriving at a customer's site to discover they don't have the one thing they need to complete the job.

Engineer Paperwork

Once all this is complete and you, hopefully, have resolved the situation, the reporting doesn't end there. The engineer will need to submit a site report. This paperwork is crucial, in fact it can be argued that it's the *most* crucial form of them all.

The engineer paperwork helps you identify several things that can feed back into your documenting, reporting, and training for both IT support personnel and for the people you support. This can include problems that may become common and that can be fixed more easily and quickly now they've been identified as such.

Additionally, it can help inform where changes to the documenting of problems should happen. This is especially useful when supporting hardware, software, or services that are new or that have been upgraded and changed.

Figure 13-6 shows an example engineer form, following on from the second-line support report for the engineer seen earlier. Here we can see that the engineer has identified a faulty network socket on the user’s premises. This information can then be passed to that building’s management and maintenance team so that the socket can be replaced.

Final Site Report

Problem found
(Detail if no problem)

All cables fine between PC and printer and USB ports okay on PC, problem was found to be the network socket the printer was also plugged into, a wiring/signalling problem was confusing the printer. When the CAT5 cable was plugged into a different socket, the problem went away.

Steps taken to resolve issue

Printer has been unplugged from network socket, and CAT5 cable left on desk with customer.

I have informed the office manager and building's maintenance that network socket E354 on the 3rd floor needs replacing.

User printing okay from USB.

Hardware / Licences used

Network socket (building maintenance)

Nothing provided by engineer

Figure 13-6. The engineer notes can feed back into the system

Additional Forms and Reports

You may find that because of your own specific needs and the needs of your customers, you will want additional reporting forms and processes. You may, as an example, want a form on which people can suggest topics to add to the staff training program or changes to existing training.

You may also want staff to be able to suggest alerts to be passed to all personnel, perhaps detailing a design flaw or a hardware fault with a product or an entire product range. All of these can prove useful, and you'll want to have a formal process for documenting and handling this information as messaging and email make it all-too easy for information to be lost or overlooked.

Summary

It's great to be able to query end users to find out what problem they're facing and what is it they might have done already to try and fix it. If you are encountering a technical issue with a PC system though, finding the information you need can be a daunting challenge.

In the next chapter, we'll begin looking at not just how you can obtain this information but at all the different types of reporting and logs available to you on a PC system, including the extremely useful Event Viewer.

CHAPTER 14

Harnessing System and Error Reporting in Windows

Microsoft Windows is an absolute treasure trove of information and data. There is literally nothing that happens on a PC without the event being recorded, and while this might sound a little scary from a data-privacy point of view (it's not as this data is only ever stored locally on the PC¹), it's just fantastic from an IT Support technician's point of view.

In this chapter I want to show you what's available in Windows, how you can access it, most importantly how you can get the very best from it, and the levels of detail that are available to you.

It's great too that this data and the reporting systems are available for data, not just that's happened in the past. Reporting can be provided immediately when a problem occurs, immediately notifying the user, yourself, or both. This can be extremely helpful for those intermittent problems that can seemingly happen at random intervals.

All of the tools I'll detail in this chapter are available in every currently supported version of the Windows operating system.

Reliability History

The first of the tools to detail is the *Reliability History*, which you can find by searching for **reliability** in the Start Menu. This is the most basic, but also one of the most useful tools available in Windows.

¹Some telemetry data, such as crashes and Blue Screens of Death are always reported through Windows Update to Microsoft.

It shows a left-right timeline view of each day for which data is available. On each of those days, it shows information events (highlighted by a blue circle containing letter “i”), warning events (highlighted by a yellow triangle), and critical events (highlighted by a red circle containing the letter “x”); see Figure 14-1. Following this is a graph, showing a score for the PC’s general reliability over the period of time displayed.

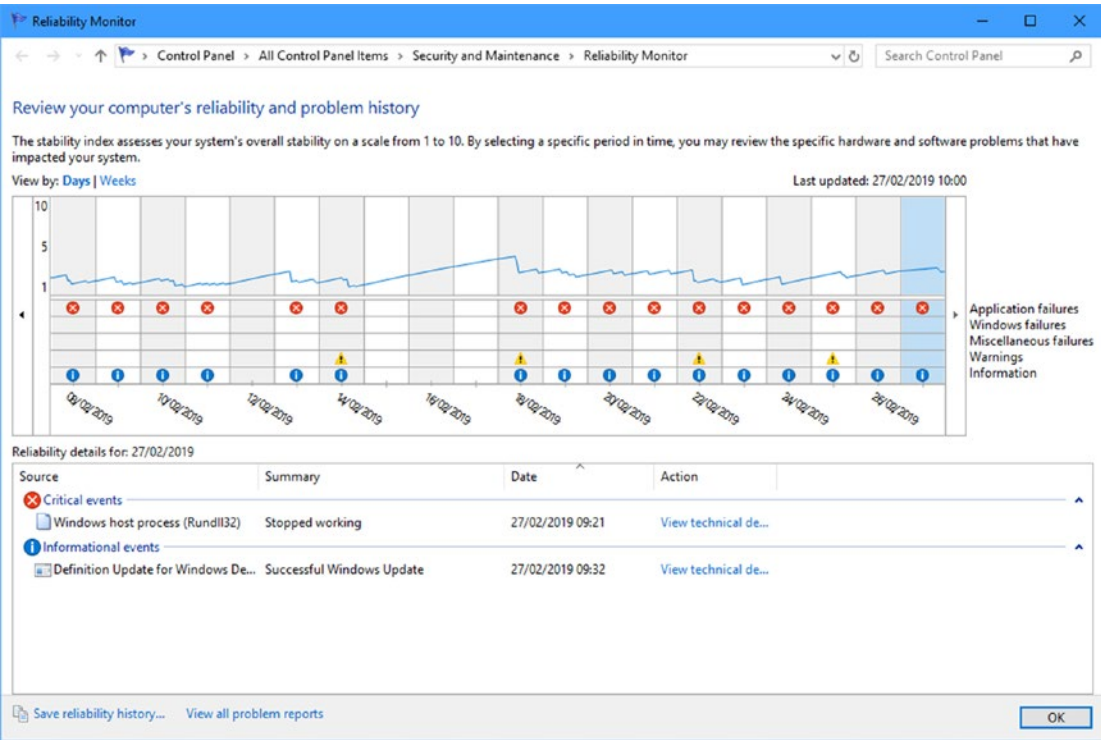


Figure 14-1. The Reliability History tool in Windows

You can click on any date in the Reliability History to view details of any information, warning, or critical events on that day. Though the information is fairly basic, just a one short sentence summary, it can often be enough to give you useful information about what has happened. For example, it might immediately be able to tell you that a specific service or driver has crashed, or that Windows Update has been unable to install updates.

Next to each event though is a *view technical details* link. Clicking this changes the view to provide much more detailed information on what has occurred. This is where the truly useful information can be found. In Figure 14-2, we have details of the exact dll (dynamic link library) file that has crashed, complete with its version number.

Other reports will give you specific Windows error codes which you can search for online to get more technical information about how to fix problems, and these always come in the format 0x00000A00.

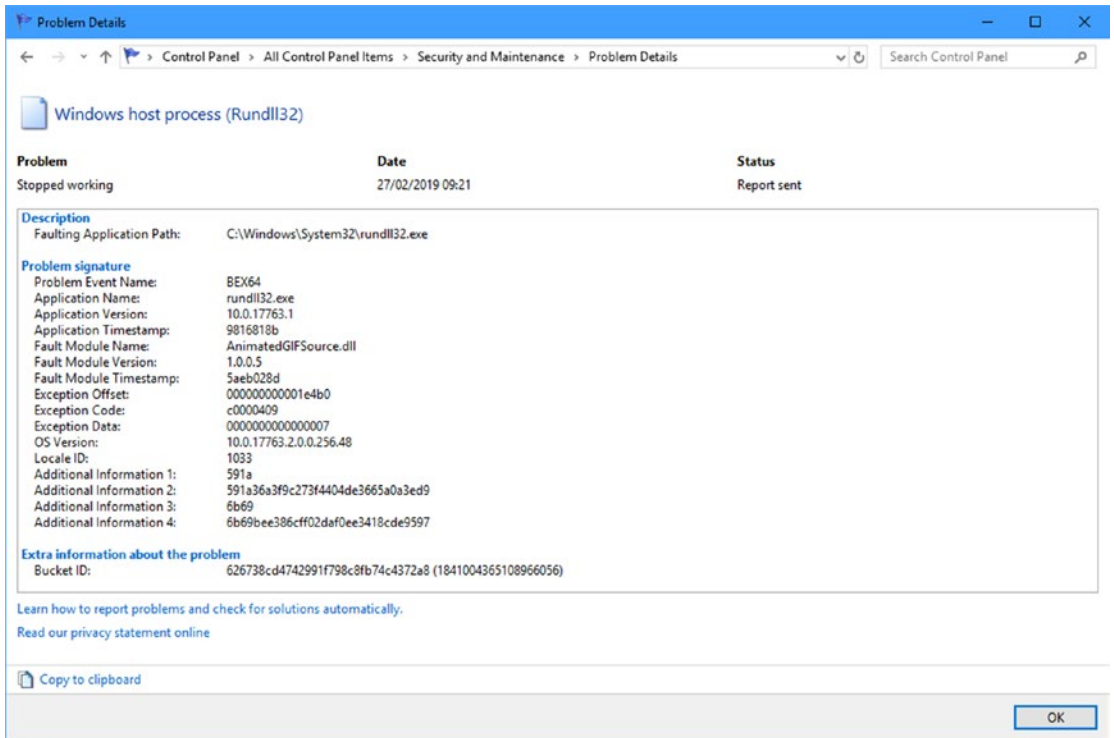


Figure 14-2. You can view technical information about events

At the bottom of the screen is a *copy to clipboard* link that you can use to cut and paste the error details, perhaps into an email to send to a colleague.

You can also save the entire Reliability History. At the main screen, you have a link at the bottom of the window to *Save [the] reliability history*, and clicking this saves the entire available history as an XML file that be perused later or sent on to another individual.

Clicking the *View all problem reports* link will also give you a full list of all errors, see Figure 14-3, and this can be useful for seeing just how often an error has occurred and whether or not it has been automatically reported to Microsoft to try and find a solution automatically.

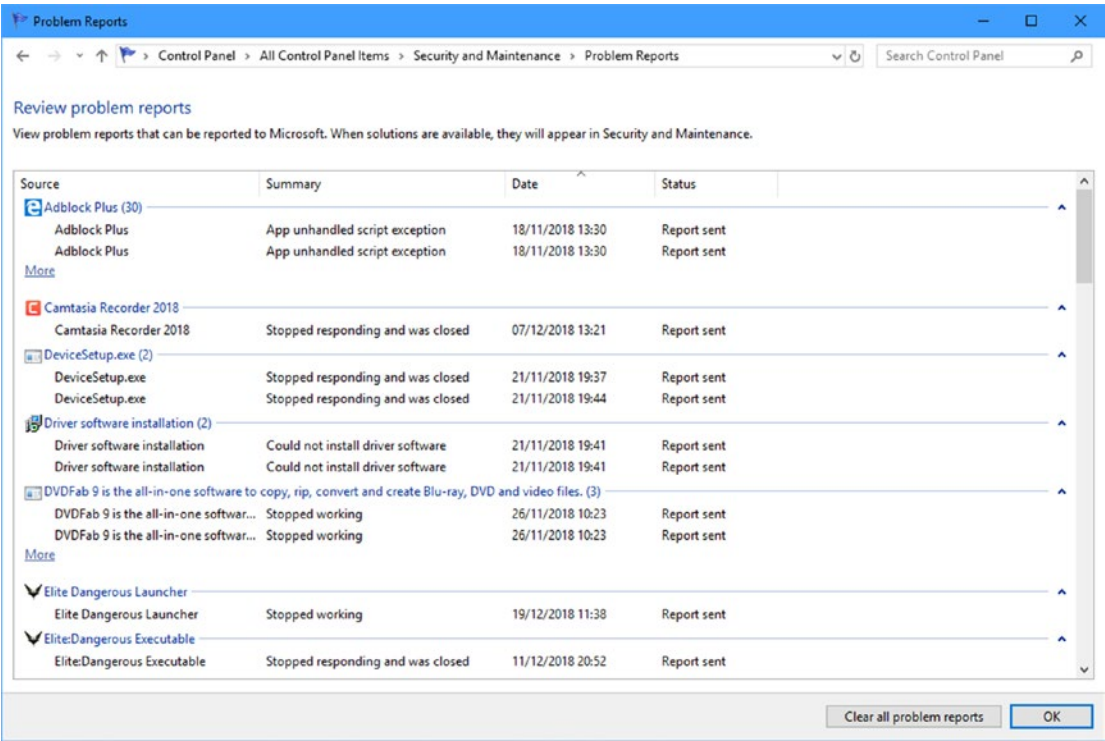


Figure 14-3. You can view all error reports on one screen

Administrative Tools

Hidden away in the Control Panel in Windows is a secondary panel called the *Administrative tools*, though often it's easier just to search for this in the Start Menu. There is a wide array of useful configuration and diagnostic tools here, including the Advanced Windows Firewall and the Computer (and Disk) Management Consoles.

There are three utilities in Administrative Tools however that are especially useful in diagnosing and troubleshooting problems on a PC, and I'll detail each one of these in the following section.

System Information

If you want to get detailed and technical information about PC, there really is only one place to go: the *System Information* tool. This tool contains data on absolutely every single element of the PC's hardware, software, and configuration; see Figure 14-4.

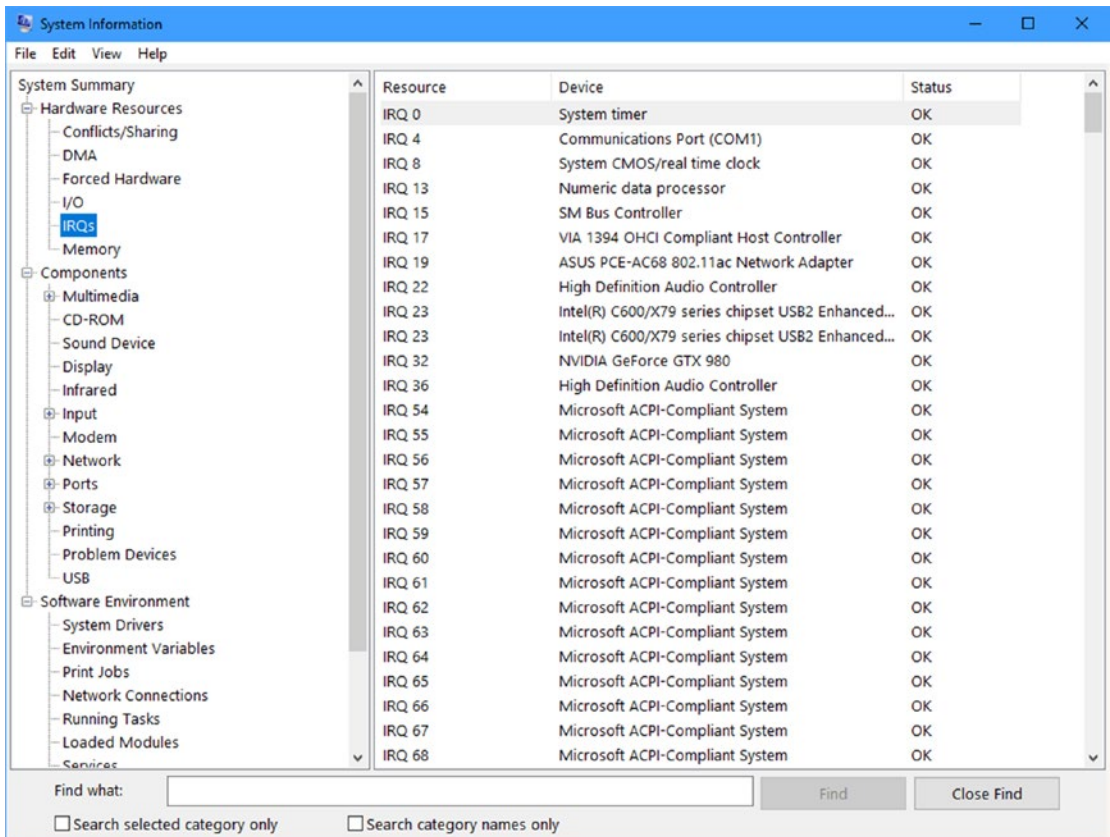


Figure 14-4. System Information provides a wealth of information about the PC

It's split into three main sections, each with sub-sections within. Hardware resources is where you can find technical information about the hardware installed in the PC, as well as details about driver conflicts, IRQs (Interrupt Request Channels to the processor), current memory usage, and more.

Components contains full details of all the hardware installed in the PC, including details of driver versions, the driver files themselves, and of any devices currently identified by Windows as having a problem.

Lastly, software environment is where you can find details on every type of software install. This includes hardware and software drivers, user (environment) variables that set preferences for their account, network connections, services, and installed problems.

There are two useful features with the System Information panel. Firstly, under the *File* menu in the top left corner of the window is the *Export* option. This allows you to export a text file containing every piece of information about the PC, which can be useful if you need to send this information to somebody or want to chew over it at another time.

Additionally, the *View* menu contains a *Remote Computer* option. This can be used to connect the System Information panel to another PC on the local or on a remote network. You can type the name and domain of the PC, or its IP address to connect, and view the full details of that remote PC.

Performance Monitor

If it's live information about a PC that's needed, then there's no better tool to use than the Performance Monitor. This displays any of thousands of live metrics about the PC and its performance graphically as a scrolling line graph, and it's highly configurable.

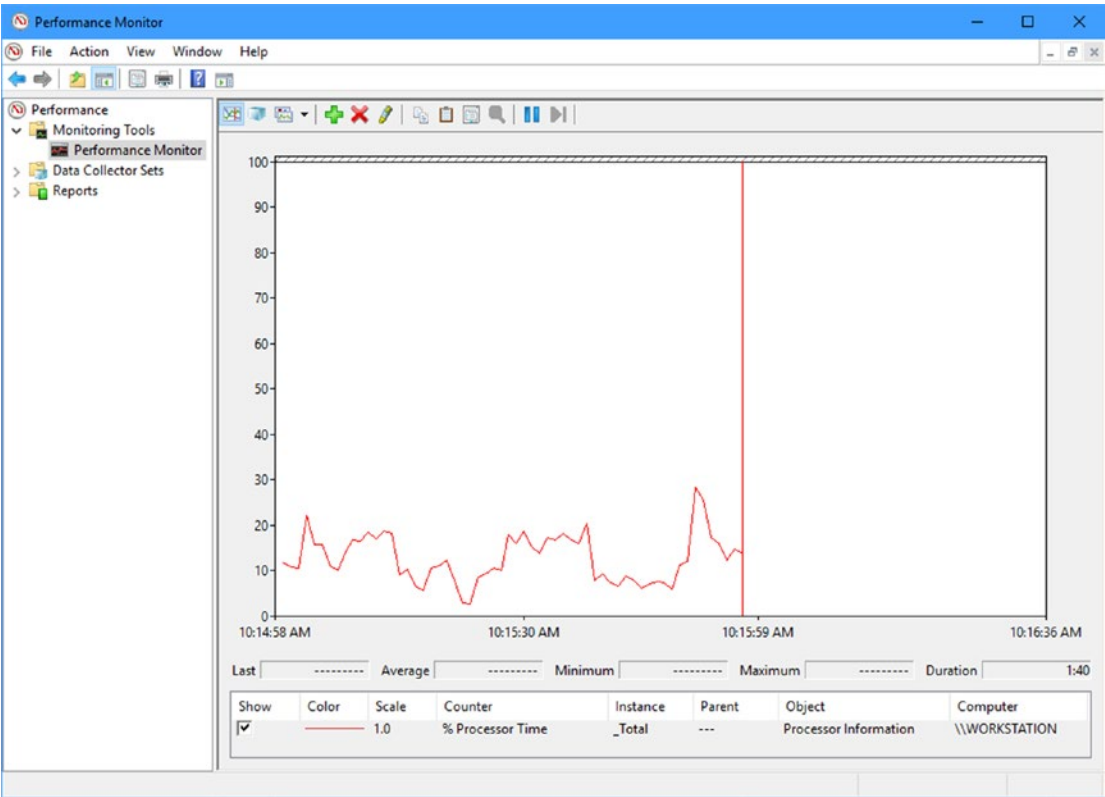


Figure 14-5. *The Performance Monitor*

When you open the Performance Monitor, it begins drawing a line graph for the performance of the process, as seen in Figure 14-5, though the toolbar allows it to be used for much more.

Along the top of the window is a toolbar. Clicking the green + symbol lets you add one or more from thousands of available metrics that you can then monitor in real time; see Figure 14-6. Select the metrics you want to monitor (you can hold the Ctrl key on the keyboard to select multiple metrics), and then click the *Add* button to add them to your active counters list.

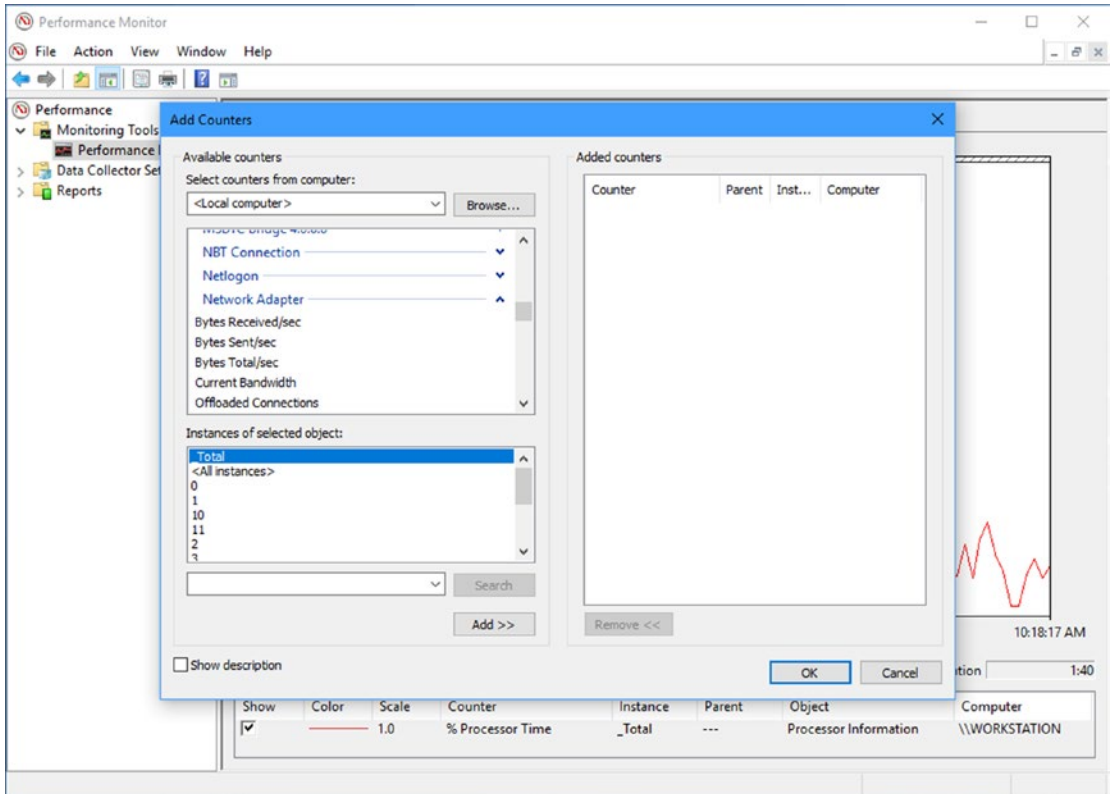


Figure 14-6. You can add graphs to the Performance Monitor

The benefit of using the Performance Monitor is that it enables you to view, in real-time, what's going on with any part of the PC's hardware or software. You can, for example, monitor the network traffic to look for dropouts (see Figure 14-7) or monitor the disk write or read metrics to help you discover what's happening on the PC when intensive operations are taking place.

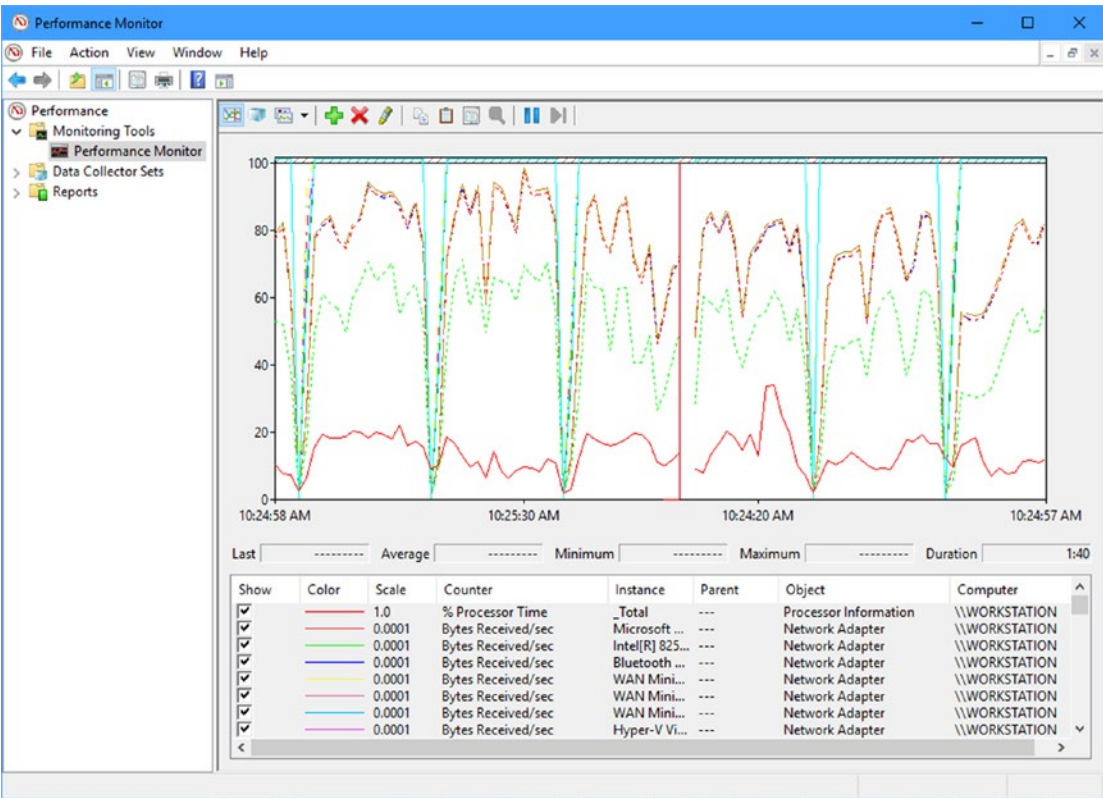


Figure 14-7. You can monitor many metrics in Performance Monitor

Click the *Highlight* icon on the toolbar and then click any one of the items you’re monitoring in the panel at the bottom of the window, and it will be immediately highlighted in the graph for you to see more easily; see Figure 14-8. This can be useful in scenarios where you are monitoring a lot of different, or related, metrics but need to quickly be able to see the data about one or more.

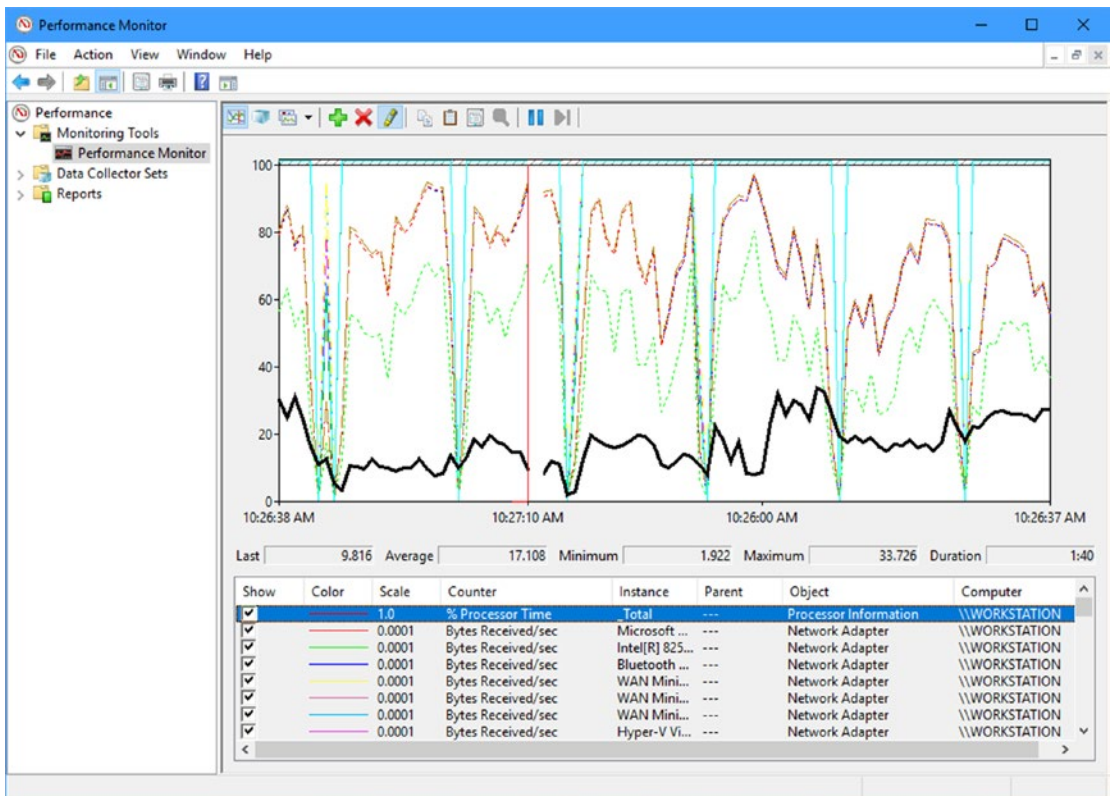


Figure 14-8. You can highlight any metric to draw attention to it

Data Collector Sets

While the Performance Monitor might be great for watching PC activity in real time, what happens if you want to collect data over a longer period? It's here that *Data Collector Sets* can come into their own. Data Collector Sets are exactly what they sound like – records of specific performance metrics recorded over a period of time.

You create a new Data Collector Set by opening the *Data Collector Sets* section at the left-side panel of the Performance Monitor and then opening the *User Defined* sub-folder. Any Data Collector Sets that are already defined will appear here, and you can create a new one by right-clicking any space and selecting *New* from the menu that appears.

You're asked to give the Data Collector Set a name and other basic parameters to create it. And then it will appear as a sub-folder in the *User Defined* section. Clicking on your new Data Collector Set, you can then right click in the blank space in the right-side panel and add a new Data Collector.

You can collect any type of data on the PC’s performance here, and you select which metrics you want to collect in the same way as entering them into the main graph; see Figure 14-9.

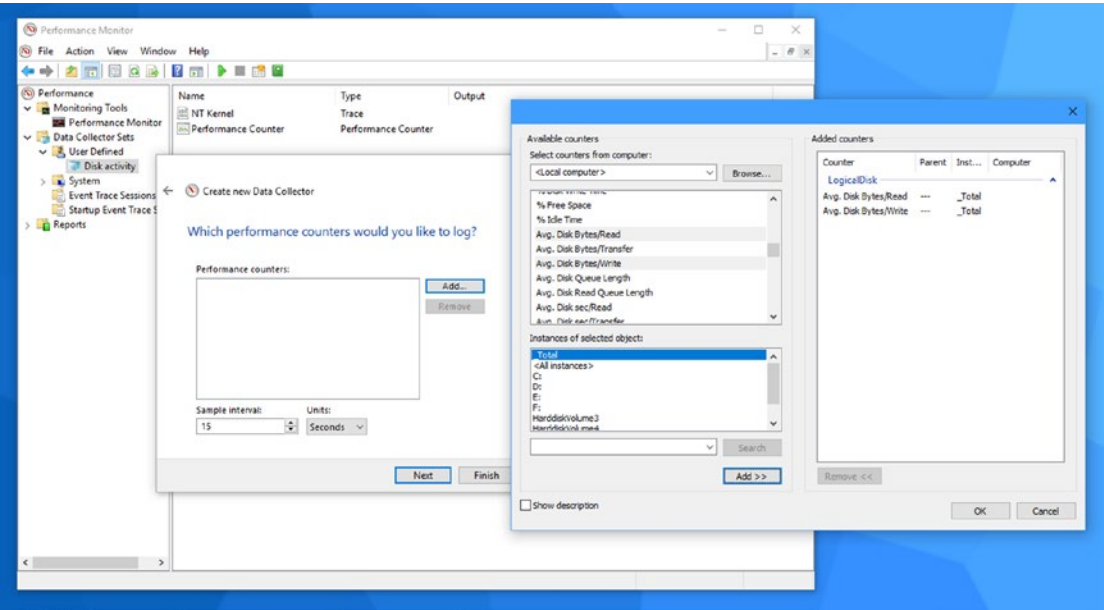


Figure 14-9. You can create Data Collector Sets to monitor performance over time

With the Data Collector Set (I feel this should be a drinking game, Ed.) created, click the green arrow icon above it on the toolbar to begin collecting data. You will be shown the directory and file name and location for the collected data, which is always stored in the `C:\PerfLogs\` folder. Opening these files enables you to see the collected data.

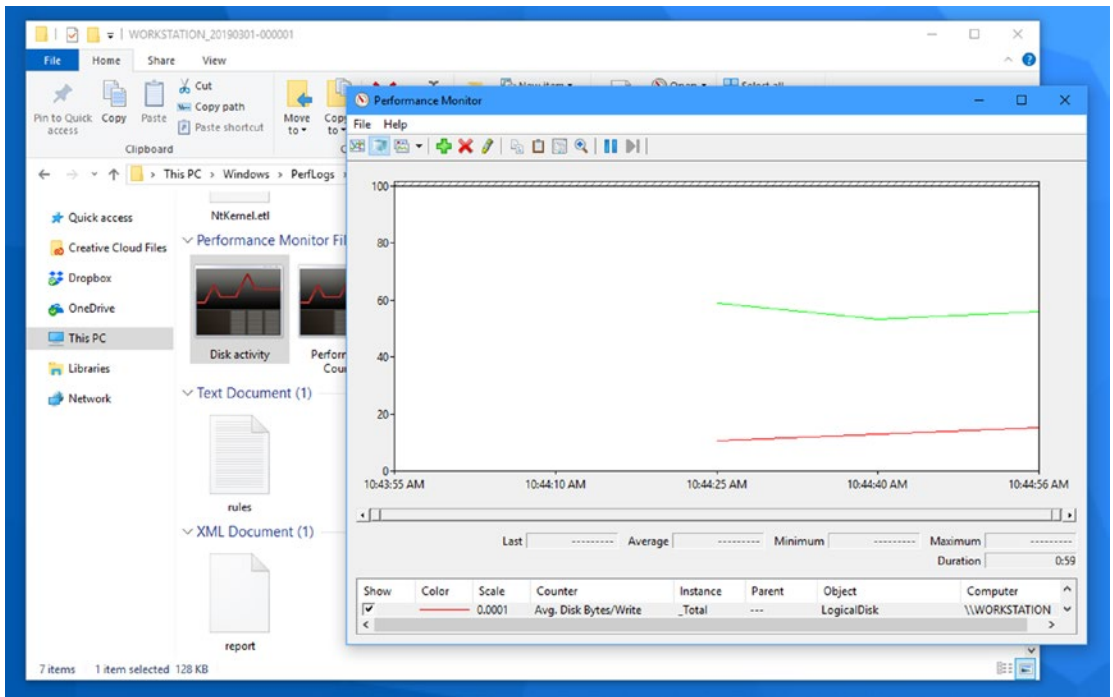


Figure 14-10. You can open your data collector set from File Explorer

Event Viewer

By far the daddy (or these days should that be significant parent) of the Windows diagnostic and reporting tools is the Event Viewer, and it's probably the first thing many IT pros will look at if there's a problem, even though other tools might provide better and more relevant data.

The Event Viewer (see Figure 14-11) is standard management console fare, with a left panel containing all the different available views, the right panel containing context-sensitive controls and options, and the center panel providing the information.

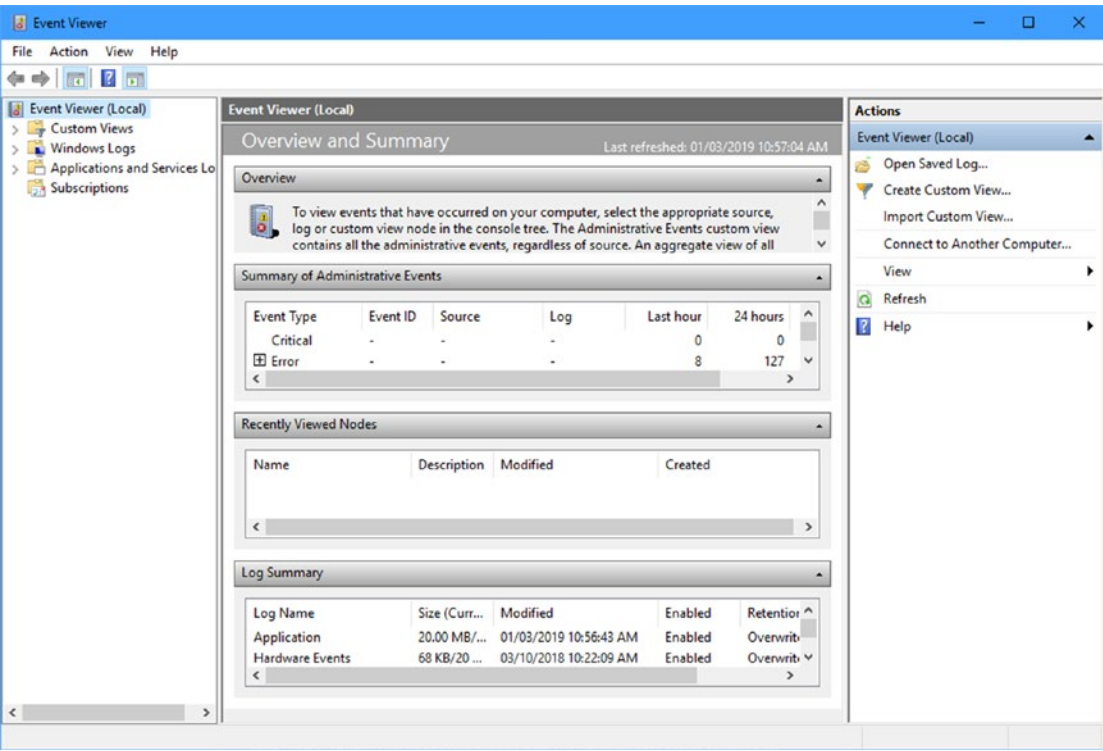


Figure 14-11. *The Windows Event Viewer*

This center panel contains four separate, collapsible sections, each able to provide different information or context on what you’re viewing. The most useful of these is the *Summary of Administrative Events*, and it’s here that you can see a list of all the *Audit successes* (basic monitor events), *Information* (records of things like when services and apps and started), *Warnings* (where something goes wrong but that doesn’t affect anything on the system), *Errors* (where something goes wrong that does, or potentially can affect the system), and *Critical Events* (such as a Blue Screen of Death).

Expanding any of these event sections will display a list of the different events that have occurred, collated by type. You can double-click any of these to display a list of every time that an event has occurred. Below this is more technical information about the event itself, including any error codes that may be relevant, and a verbose description of what occurred, see Figure 14-12.

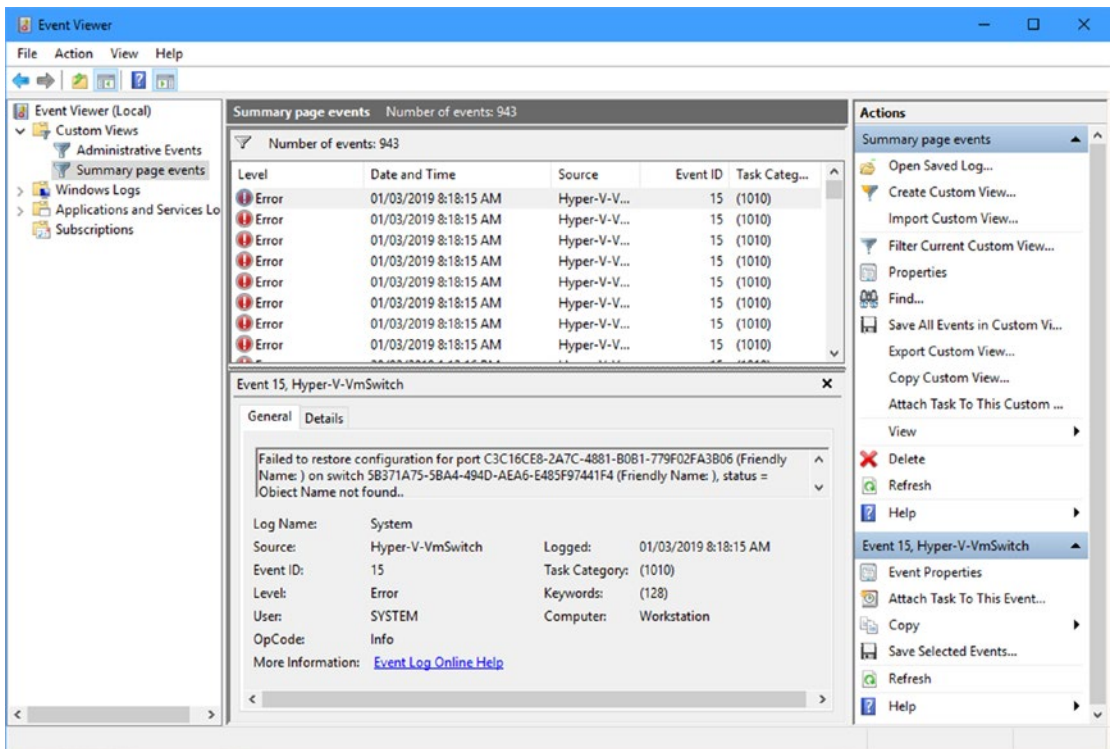


Figure 14-12. You can get detailed information about events

Each panel in the Event Viewer contains sortable column headers, which enables you to order and view the events by when they occurred. This can help you narrow down what was happening at the time the user reported an error.

In the left panel is also a *Windows Logs* section. It is here you will find logs specifically recording application, security, setup, and system events. These sub-sections can also make it straightforward to find the information you're looking for.

Creating Custom Views

One of the issues with the Event Viewer is that it stores details on absolutely everything that happens on a PC. While this can be incredibly helpful, it can also be extremely difficult to sift through, in order to find the information you need.

It's this reason that the Event Viewer allows you to filter available information, so that you only see what you want and need to see at any one time. To filter the logs, open the view you wish to obtain information about and click the *Filter current log* link in the right-side panel.

A dialog pops-up asking how you want the view filtered; see Figure 14-13. Many options are available to you here, from only viewing events of a certain type (critical, error, etc.) to only viewing events related to particular features of Windows (such as AppV).

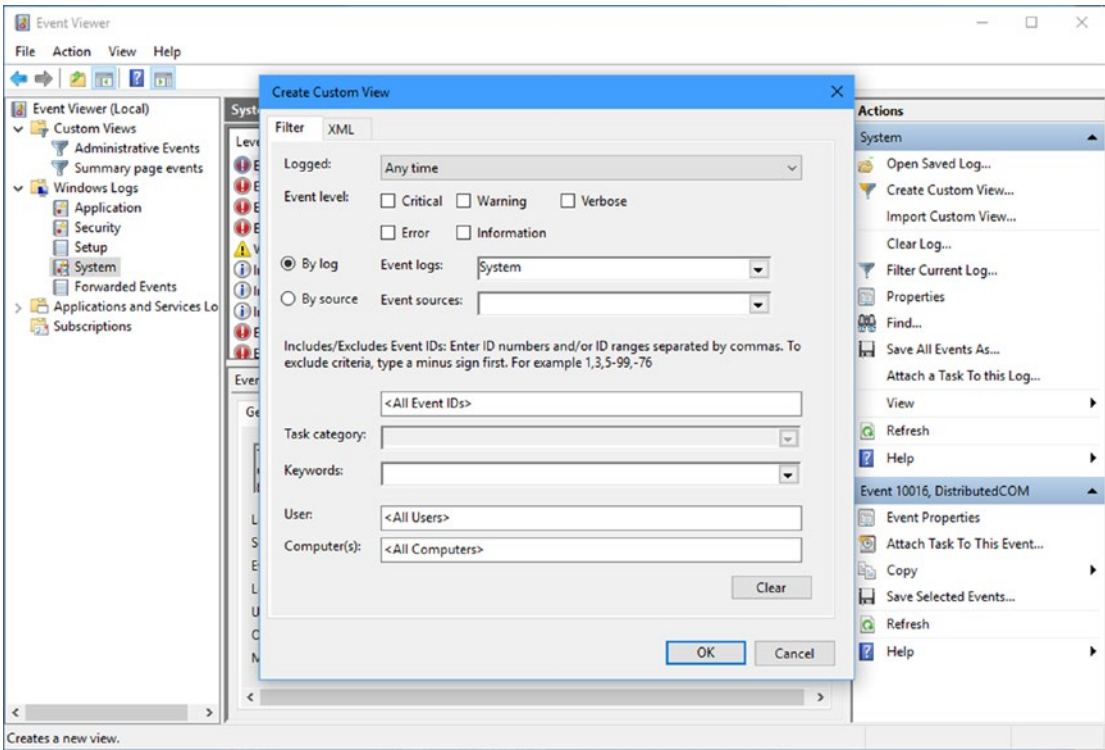


Figure 14-13. You can filter the event log

Perhaps where filtering comes into its own is with the *Event sources* drop-down menu. Here you can select only the Windows services and features to present data for. If, for example, you are having trouble with the network connections, then you can choose to view only one or several of the specific network events that are recorded.

You can additionally choose the view of the data for all users on that PC or, if more than one person has an account, one or several of those users. The way you can obtain just and only the information you require is very impressive.

Attaching Tasks to Events

There are times when you want to be alerted straight away when an event occurs. Let's say that a problem a user faces is intermittent, and you need to know exactly what they are doing at that specific time. If this is the case, then Windows can alert you automatically.

Now I need to caveat this with a notice that this feature is being deprecated in Windows 10. There's no firm details as I write this about exactly when it will be removed from the operating system and what Microsoft will replace it with (keep an eye on my web site www.windows.do and I'll write more about it when the time comes). The feature will remain in Windows 7 and Windows 8.1 however.

When you have found the event you wish to be notified about, highlight it and in the right-side panel click the *Attach a task...* option. This displays a dialog that allows you to start a program, send an email, or display a message (see Figure 14-14).

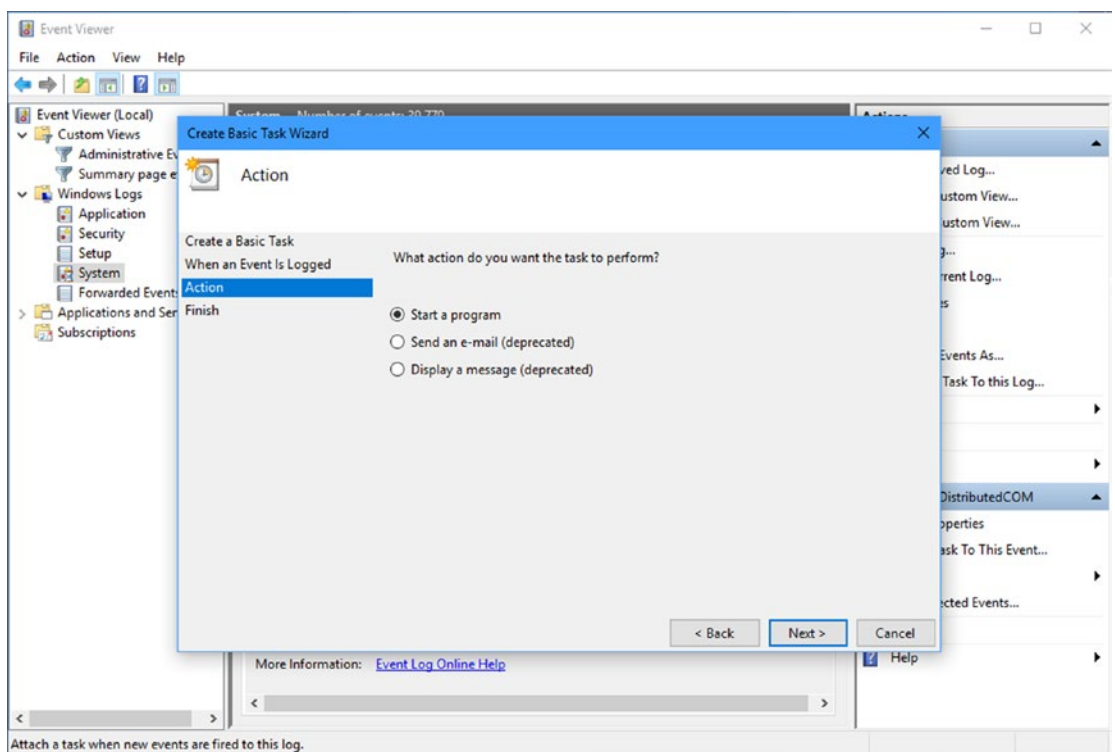


Figure 14-14. You can attach tasks to events

This can be useful in several ways. You can, as an example, have a pop-up alert appear on the user's screen that can inform them that they need to immediately stop what they're doing and call IT support, so they can detail exactly what they were doing when the error occurred, what's running on the system, and any other information that will be relevant to the diagnosing and troubleshooting the error.

Getting More Use from Event Data

There are various ways to save details about events, so that they can be read later, or sent to somebody else for analysis. You can highlight one or more specific events and then click the *Save selected events* link in the right-side panel, or for the currently selected view, and again in the right-side panel, you can click the *Save all events as* link.

Additionally, in the main Event Viewer panel (click *Event Viewer [local]*) in the left-side panel, you can click the *Connect to another computer* link to connect to and view the event logs on another computer connected to your computer or to which you have access rights online.

Honorable Mention: Task Manager

As an honorable mention, and this doesn't apply to Windows 7 which has an older version of the tool, the Task Manager can be an incredibly useful tool for monitoring a system. I want to look at some specific examples.

The first of these is the *Processes* tab. This allows you to see live data about the CPU, memory, disk, and network but in a way that is heat-mapped. This highlights any moderate or heavy usage in orange or red, making it straightforward to spot offending processes.

The *Performance* tab displays live graphs about various operations on the PC, such as disk usage or network traffic. However it's actually much more useful than this. You can right-click (or double-click) any of the graphs and view them in different ways. This includes being able to view the data numerically, instead of as a graph, and to get mini-views of the different metrics available, and also of an individual component (see Figure 14-15).

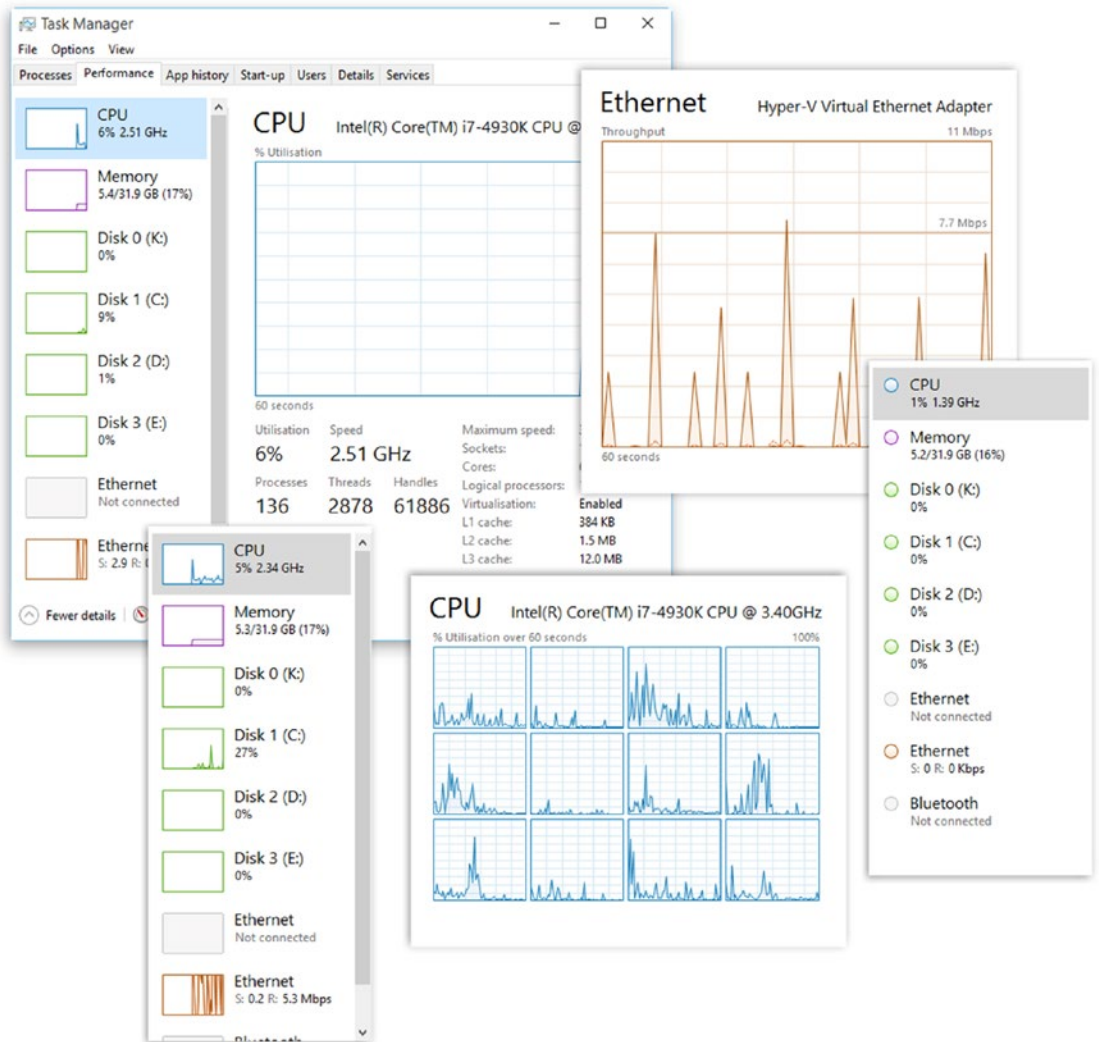


Figure 14-15. *The Task Manager is highly configurable*

The upshot of this is that if you want to be able to keep a close eye on something, but don't have a lot of desktop real estate on which to have a Windows open such as the Resource Monitor, you can reduce the Task Manager to one of these mini-views and stick it in a corner of the screen, out of the way.

Summary

Windows contains a wealth of reporting and monitoring tools, and it stores details on absolutely everything that happens on a PC. This is all very useful, but what happens if you need even more information from the system than these tools provide, such as details of past Blue Screens of Death? Well how you obtain that and other detailed information is what we'll cover in the next chapter.

CHAPTER 15

Obtaining Advanced Error and Status Information on PCs

So far we've only looked at the Windows Event Viewer in an overview. What it's capable of however is much more than just storing error codes for blue screens and crashes, so let's look at the Event Viewer in more detail.

Getting Detailed Information About Errors

When you click on an error in the Event Viewer, you're given a wealth of information about it; see Figure 15-1. This begins in the top-central panel where you are shown each and every recorded instance of the error or event. I say *recorded* because Windows occasionally clears out old logs when it feels they'll no longer be needed, which is usually after 30 days. They can also be cleared manually though through the use of a tool, like Disk Cleanup, or the popular third-party CCleaner.

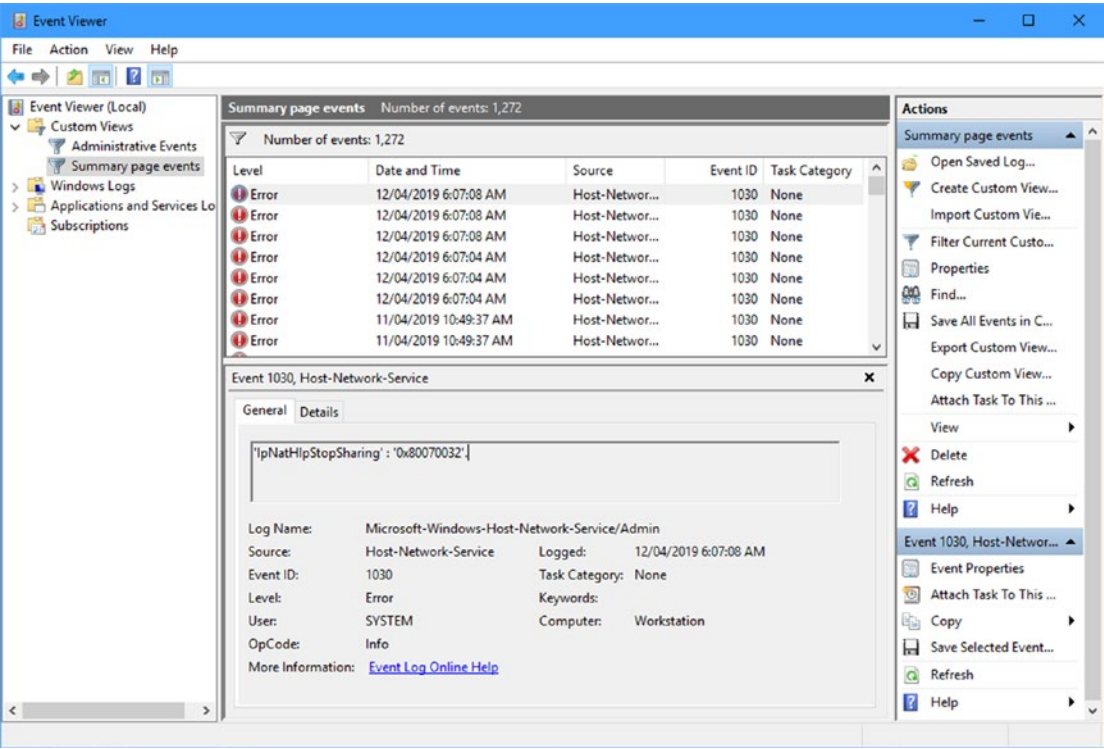


Figure 15-1. The Event Viewer

The error details found in the Event Viewer include the full date and time of each instance, and this can make it simple to search for all errors and events around a certain time. You can achieve this in the main Event Viewer window by clicking *Create Custom View* in the right-side panel, and from the *Logged* drop-down, selecting the time period you wish to view events for; see Figure 15-2. This includes timeframes from the last hour to the last 30 days but also gives you a *Custom Range* option.

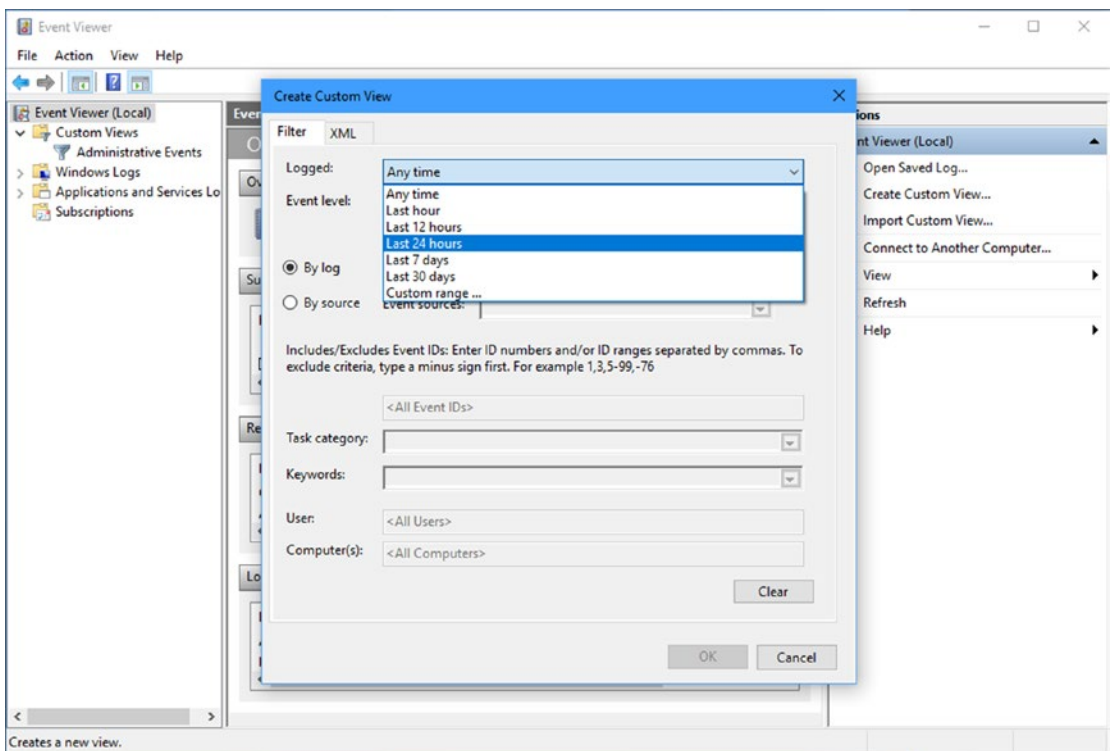


Figure 15-2. You can view events by time and date

Additionally, and as I mentioned in Chapter 14, you can create *Custom Views* that allow you to further filter the events by their type and severity. Clicking an event provides further information about it in the center-bottom panel; see Figure 15-3. The Event ID can be used to search in the Event log or online for similar events, or solutions to that event. The source tells you which Windows component, app, or driver has encountered the error, and in the box above this information is a verbose description.

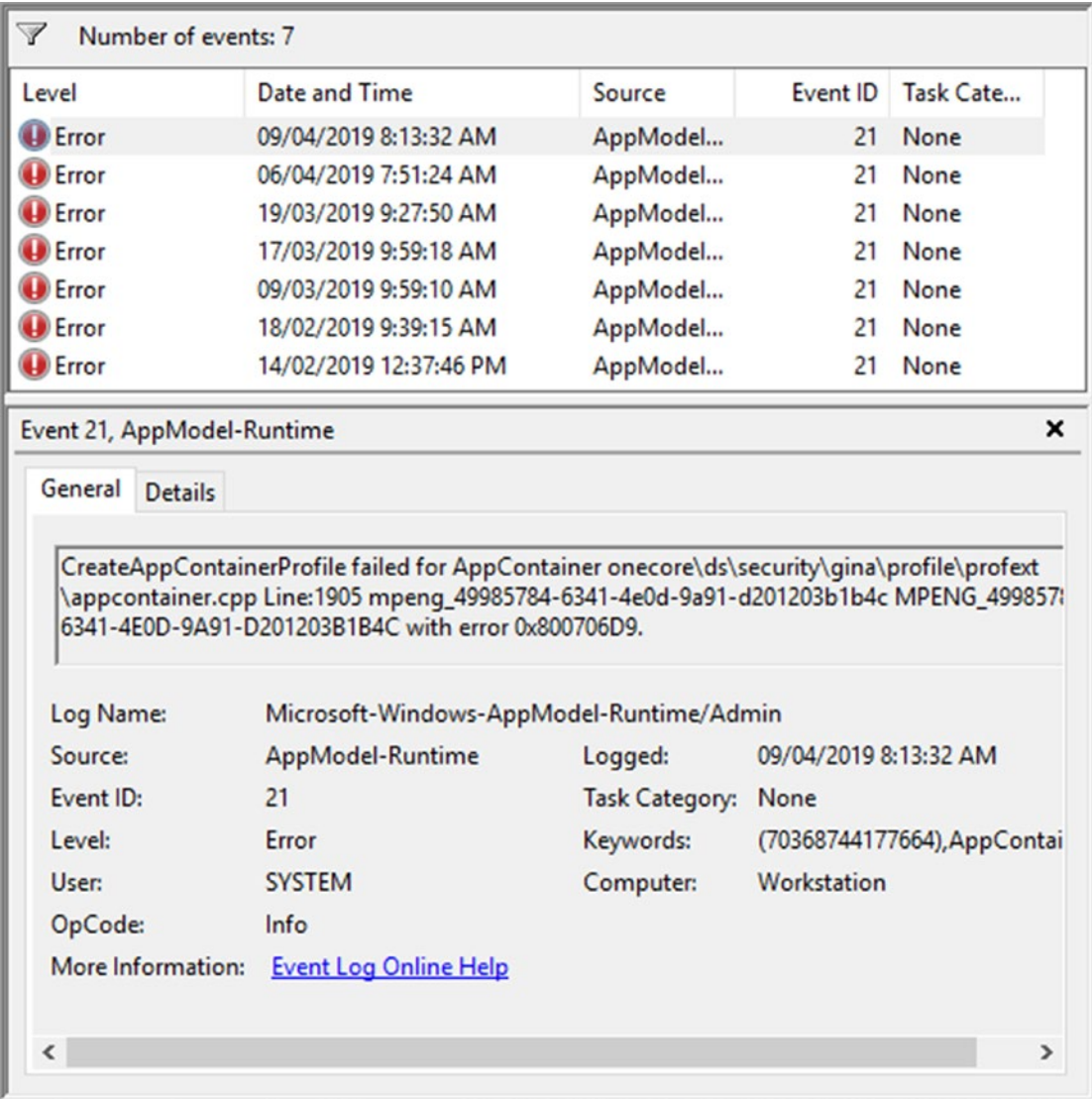


Figure 15-3. You can get detailed information about events

This description will often feature a Windows error code. These are always in the format 0x00000000 where 0x indicates it's an error, and the rest is a hexadecimal number you can search for online to provide information. In the example in Figure 15-1, we have the error code 0x800706D9 which is an error generated by Windows Update being unable to install an update or patch.

Copying and Saving Event Details

Clicking the *Event Properties* link that appears in the bottom-right panel when an event is highlighted opens a dialog with all the details about that event neatly organized. On the face of things, this isn't any different from the standard view, except that it includes a handy *Copy* button in the bottom-left corner; see Figure 15-4. This allows you to paste, in XML format, that log into a document, email, or messaging window so it can be read later by yourself or another support person.

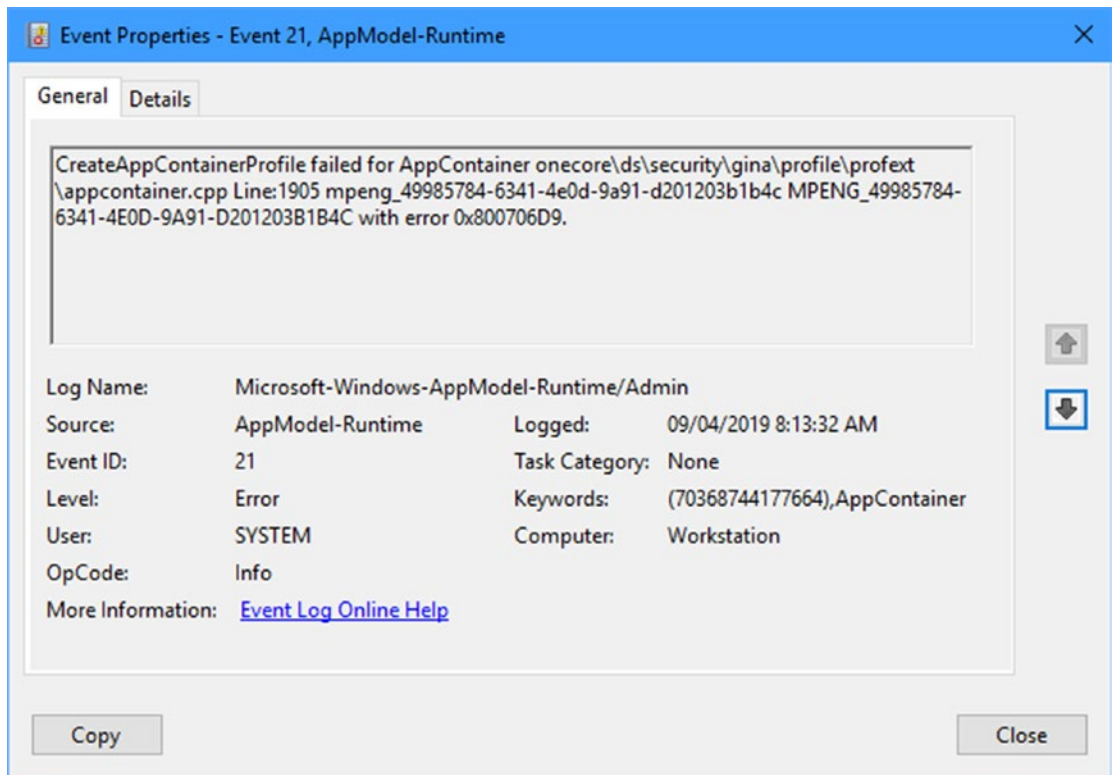


Figure 15-4. You can copy the contents of event logs

Additionally in the *Action* menu in the top-left corner of the Event Viewer window, you will find a *Save Selected Events* link. You can use this to save one, several, or all events to a .evtx file, or in several other formats such as XML. Evtx files can then be opened in the Event Viewer on any other Windows PC by clicking the *Open Saved Log* link at the top-right corner of the Event Viewer window.

Connect to the Event Log on Another PC

It's also possible to connect to the Event log of any PC that's accessible on your local or remote network. To achieve this, right-click *Event Viewer (Local)*. This displays a network connection dialog in which you can either type the network address and name of the computer you wish to connect to or browse for the PC on your network; see Figure 15-5.

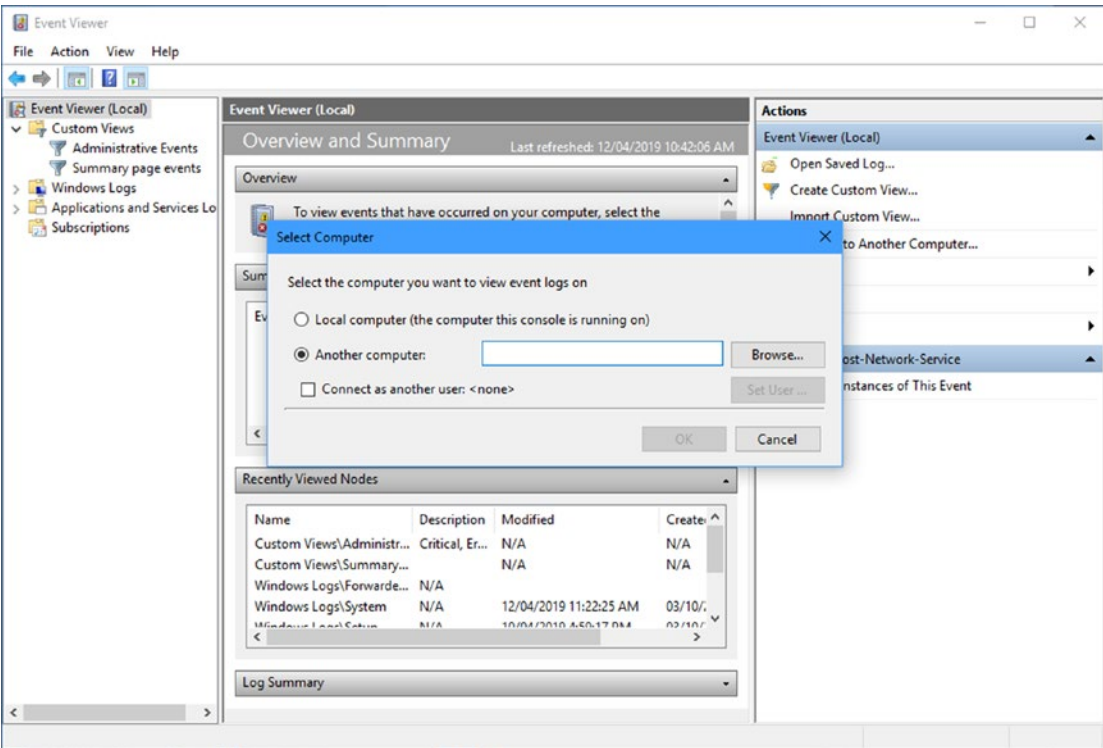


Figure 15-5. You can connect to remote PCs

Additionally you can check the *Connect as another user* check box, and you will be asked for the credentials of the account you wish to connect with. This can be very useful in some circumstances, as just as the Windows Registry has separate Registry files for each user on the PC, so the Event Viewer stores logs for individual user sessions.

Finding Other Windows Error Logs

There are several Windows log files scattered across the hard disk of a PC that you also might find useful from time to time. These contain everything from installation and updating logs to essential Blue Screen of Death information.

Text File Logs

There are plain text log files stored in the *Windows \ Debug* folder. The log files you will find here will vary depending on your hardware and software installation, and in Figure 15-6, we can see a NetSetup log with details of domain joining authentication.

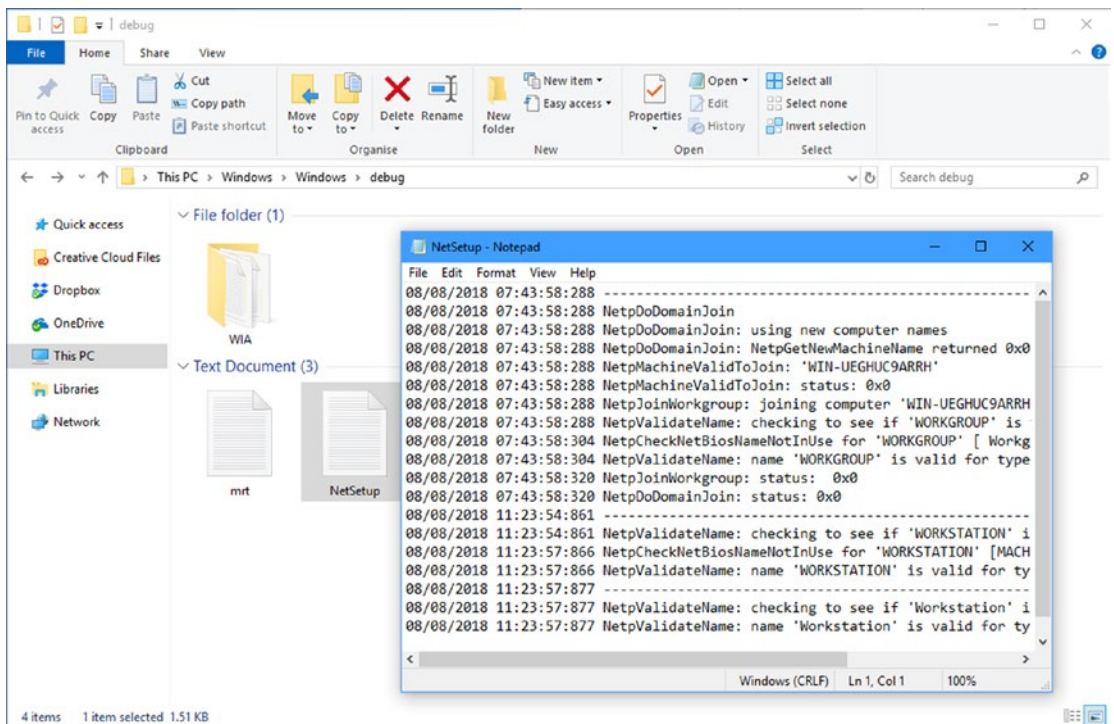


Figure 15-6. Windows stores some logs as plain text

XML and ETL Log Files

In the *Windows \ Logs* folder, you will find a series of sub-folders that contain .xml (Extensible Markup Language) and .etl (Event Trace Log) files, as well as more plain .txt files. You can open .xml files in any web browser, and .etl files can be opened from within the Event Viewer.

Again, the log files you find here (see Figure 15-7) will vary and are dependent on the software, hardware, and configuration on your PC. Etl files should also be available already in the Event Viewer.

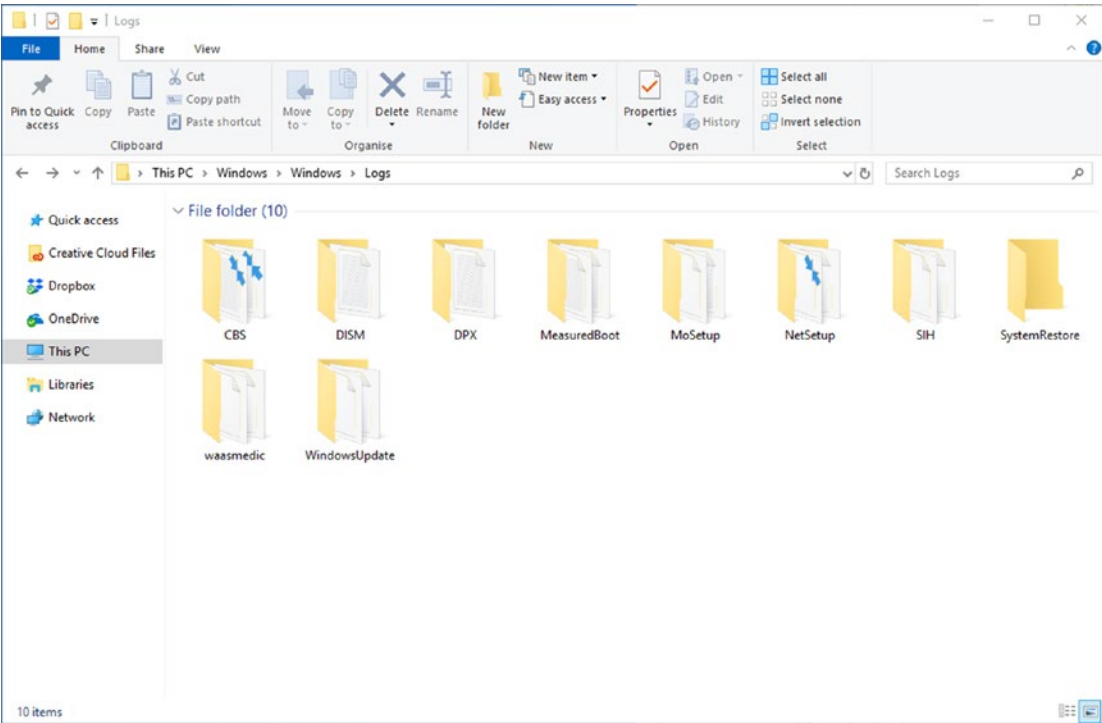


Figure 15-7. The Windows Logs folder contains more log files

Dmp Files

The *Windows \ Minidump* folder is where you will find any logs associated with a Blue Screen of Death. These are collected when, during a Blue Screen, you see a message telling you that Windows is collecting information about what happened.

You can't open .dmp files through Notepad or the Event Viewer on a PC, but there are a couple of ways to access them. If you use Microsoft Visual Studio, you can open and read the contents of .dmp files using the Windows Driver Kit (WDK) or the Windows Software Development Kit (SDK).

What can be more useful though is a third-party tool called BlueScreenView. You can download this utility from www.nirsoft.net/utils/blue_screen_view.html and use it to read the .dmp file contents so that you can see what happened during the Blue Screen of Death; see Figure 15-8.

The screenshot shows the BlueScreenView application window. The title bar reads "BlueScreenView - C:\WINDOWS\Minidump". The menu bar includes "File", "Edit", "View", "Options", and "Help". Below the menu is a toolbar with icons for file operations. The main window is divided into two panes. The top pane lists dump files with columns: "Dump File", "Crash Time", "Bug Check String", "Bug Check Code", "Parameter 1", "Parameter 2", and "Parameter 3". Two files are listed: "042416-10750-01.dmp" (crashed on 24/04/2016 at 14:14:14 with error CRITICAL_PROCESS_DIED) and "031116-17578-01.dmp" (crashed on 11/03/2016 at 3:33:53 with error CRITICAL_PROCESS_DIED). The bottom pane displays a memory dump table with columns: "Filename", "Address In St...", "From Address", "To Address", "Size", "Time Stamp", and a column for line numbers. The first row of the table is highlighted in red and shows "ntoskrnl.exe" at address "ntoskrnl.exe+640ca4". Other rows list various system files like hal.dll, kdcom.dll, mcupdate.dll, werkernel.sys, CLFS.SYS, tm.sys, PSHED.dll, BOOTVID.dll, cmimcext.sys, and ntosext.sys. At the bottom of the window, it says "2 Crashes, 1 Selected" and "NirSoft Freeware. http://www.nirsoft.net".

Dump File	Crash Time	Bug Check String	Bug Check Code	Parameter 1	Parameter 2	Parameter 3
042416-10750-01.dmp	24/04/2016 14:14:14...	CRITICAL_PROCESS_DIED	0x000000ef	ffffe000'a8b9e7c0	00000000'00000000...	00000000'00000000...
031116-17578-01.dmp	11/03/2016 3:33:53...	CRITICAL_PROCESS_DIED	0x000000ef	ffffe001'537f97c0	00000000'00000000...	00000000'00000000...

Filename	Address In St...	From Address	To Address	Size	Time Stamp	
ntoskrnl.exe	ntoskrnl.exe+640ca4	fffff801'91019000	fffff801'917e5000	0x007cc000	0x56fa1e56	2
hal.dll		fffff801'917e5000	fffff801'91858000	0x00073000	0x5632d17a	3
kdcom.dll		fffff801'8fdbb000	fffff801'8fdbb000	0x0000b000	0x5632d938	3
mcupdate.dll		fffff800'edff0000	fffff800'ee07e000	0x0008e000	0x5632d912	3
werkernel.sys		fffff800'ee080000	fffff800'ee090000	0x00010000	0x5632d90b	3
CLFS.SYS		fffff800'ee090000	fffff800'ee0f5000	0x00065000	0x5632d172	3
tm.sys		fffff800'ee100000	fffff800'ee125000	0x00025000	0x5632d166	3
PSHED.dll		fffff800'ee130000	fffff800'ee147000	0x00017000	0x5632d16c	3
BOOTVID.dll		fffff800'ee150000	fffff800'ee15b000	0x0000b000	0x5632d165	3
cmimcext.sys		fffff800'ee160000	fffff800'ee16e000	0x0000e000	0x5632d166	3
ntosext.sys		fffff800'ee170000	fffff800'ee17c000	0x0000c000	0x5632d165	3

Figure 15-8. BlueScreenView enables you to read .dmp files

Summary

It's possible to get a great deal of very detailed information of just about anything and everything that happens on a PC, or that is installed or configured on a PC. This information is also available locally or, in the case of the Event Viewer, remotely on the computer as well.

But what if you do need to provide support remotely, without any access to the PC at all? There is a wide variety of tools available, some of which will surprise you, and in the next chapter, we'll look at everything that's available.

PART V

Providing Remote Support

CHAPTER 16

Remote Support Tools

Being able to support computer users locally can be a godsend. They can give you valuable information, point at things that aren't working, and be reactive when you think of something else to ask, and you can show them what went wrong so that in the future they can either fix a recurrence themselves or, better still, prevent the problem from recurring in the first place.

There's a downside to having the user with you however. They can hover over your shoulder, ask what you're doing all the time, and just generally be quite annoying. It's not that they mean to, it's that their computer is a precious thing to them (I know, it's ridiculous really) and they want to know what's going on.

Remote support can be a better alternative, but it's also the default option for most computer support. The downside is that you have to gain access to IT systems remotely, which can prove a pain, especially if the user is in an environment with a poor broadband connection, has limited battery remaining, or is in a hurry. There are a variety of tools available though, so let's spend some time examining the various options available to you.

Remote Desktop

Remote Desktop is the daddy of all remote access tools. It is provided with all copies of Windows except the Home editions, and it can give a remote user full, unfettered access to a PC, being able to sign in as any user on that system; see Figure 16-1. Remote Desktop is a very common tool of choice for system administrators to use on their own networks.

The downside to Remote Desktop is that, unlike many other remote access tools, it's strictly Windows-only, the sole exception being able to use it to access Linux-based virtual machines running on Microsoft's own Azure cloud computing platform.

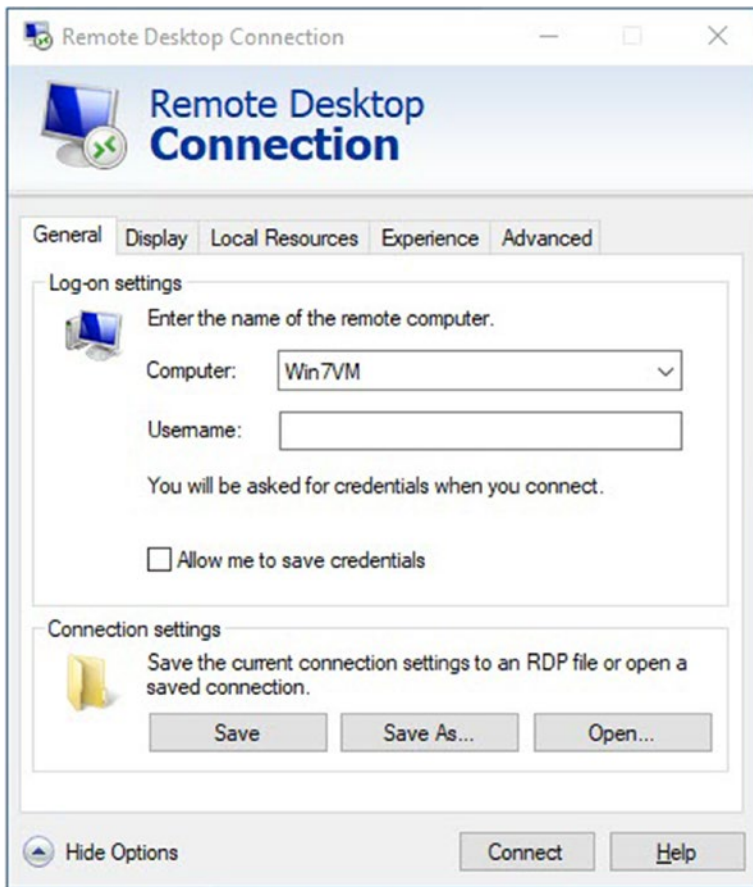


Figure 16-1. *Remote Desktop comes with all supported Windows versions*

One of the places where Remote Desktop excels is being able to sign in using the credentials of a user who isn't currently sat at the PC. If you need to examine log files or Registry entries for a specific user, the job is made significantly easier. Sure you can remotely sign into the Registry as another user; we'll look at this in Chapter 17, but it's much simpler to just take full control of their account.

One of the most useful features of Remote Desktop is the ability to share local resources on your own PC. This means if you have a drive containing backup copies of driver files, or installers, or if you want to be able to cut and paste text or other data between the two systems, it all becomes very easy; see Figure 16-2.

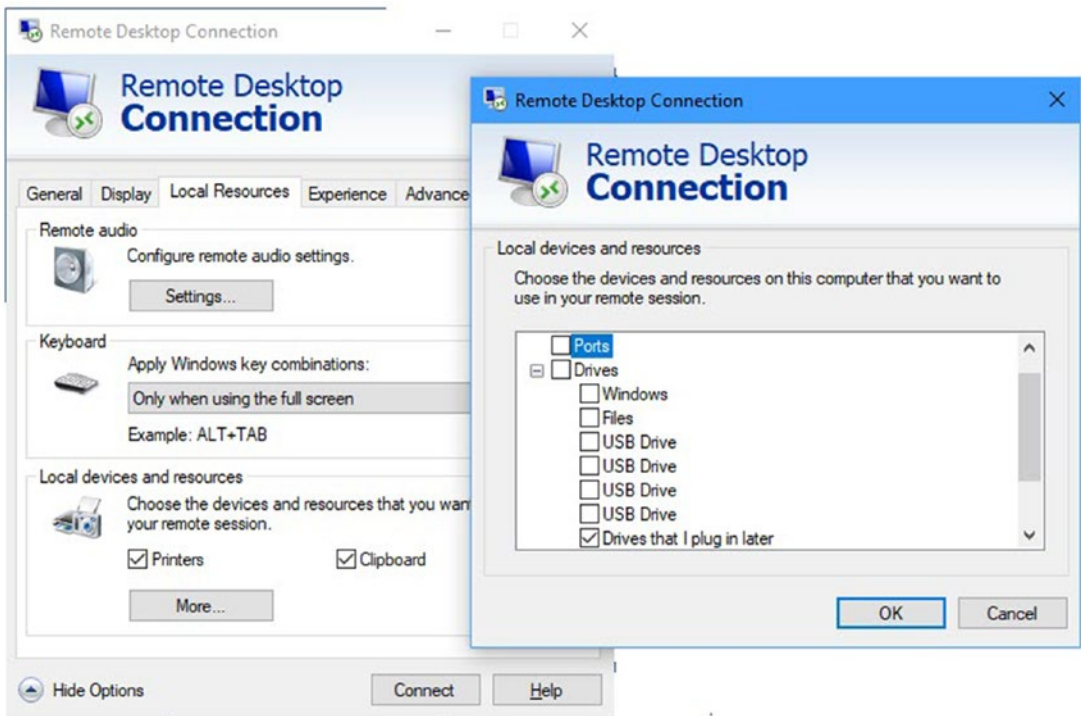


Figure 16-2. You can share local resources using Remote Desktop

Windows Remote Assistance

If you need to support Windows 7 or 8.1 PC running the Home edition of the OS, then Windows Remote Assistance could be the tool for you; see Figure 16-3. Also bundled with Windows 10, though we can expect it to be deprecated and removed at some point in the future, it provides a basic remote control experience including text-based chat. You will find this tool in the Start Menu.

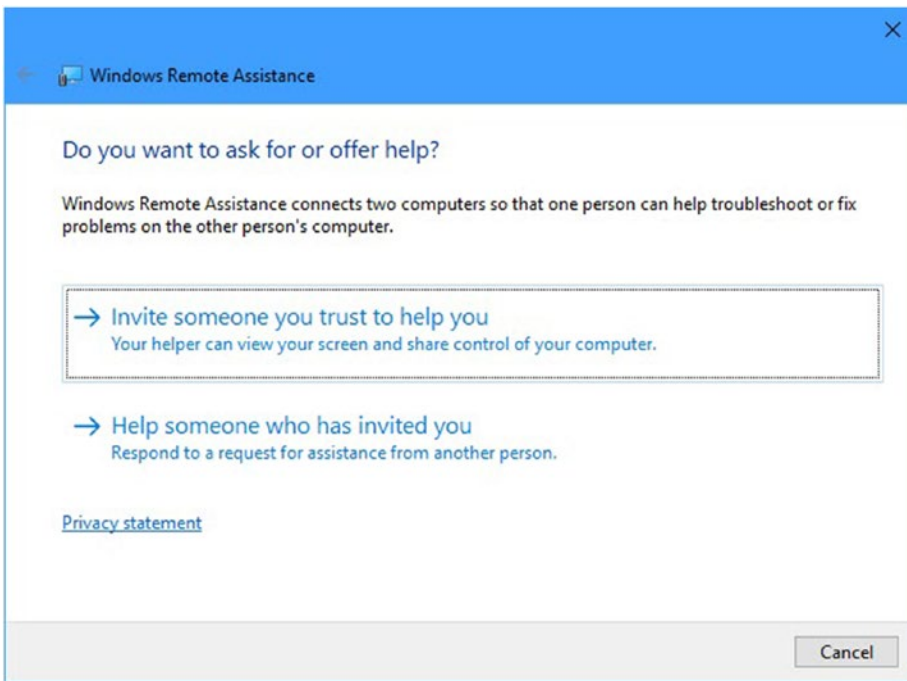


Figure 16-3. *Windows Remote Assistance is in all supported versions of the OS*

Where Windows Remote Assistance falls down is that it requires the user at the other end to be present and able to assist with configuration. Because Microsoft wants to maintain high levels of security for the user receiving support, the person providing the support must request control of the remote PC once the session is in progress. They also have to ask the end user to check a box giving them control of User Account Control (UAC) security prompts, and this being a small check box, it's easy to miss.

The biggest problem with Windows Remote Assistance however is that the session won't survive a reboot. This means that you do need to restart a PC for any reason while providing support; you have to reinstate the whole remote session from scratch.

Quick Assist

Something that *will* keep the support session going after a reboot however is Windows Quick Assist; see Figure 16-4. There are downsides however as this tool only comes with, and only supports connections between PCs running Windows 10, and you can find it in the Start Menu, though it does support the Home edition.

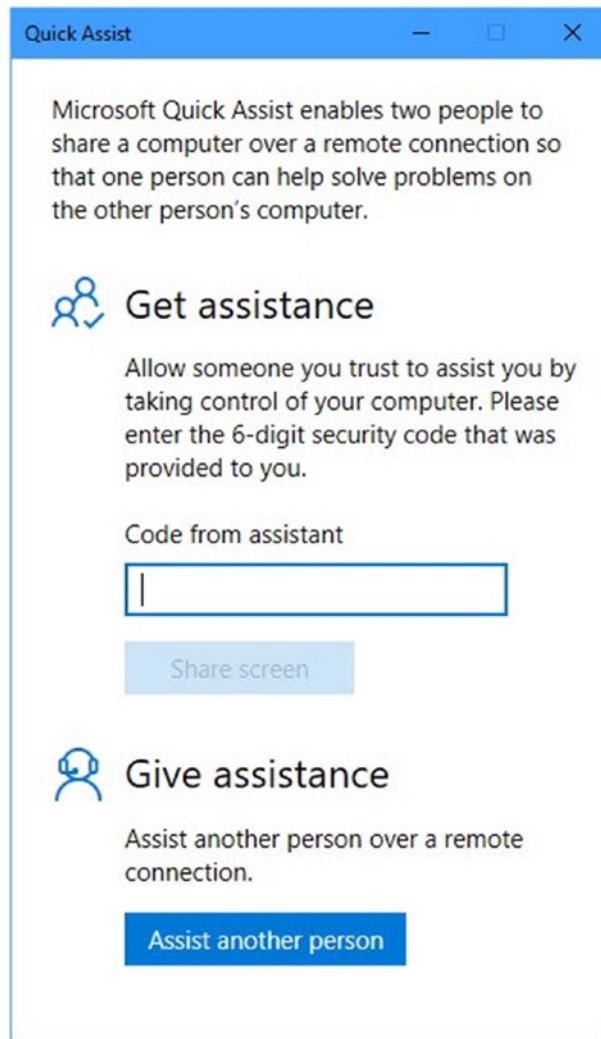


Figure 16-4. *Quick Assist only comes with Windows 10*

Quick Assist supports all of the functions of Windows Remote Assistance, including text-based chat. Additionally it supports inking, allowing for on-screen annotation.

TeamViewer

TeamViewer from www.teamviewer.com is for many people the remote access tool of choice; see Figure 16-5. It's available for Windows (both as a win32 installer and as a Microsoft Store app), Mac OS X, iOS, Android, Chrome OS, and Linux. There are also free and paid-for versions, with the latter coming with all manner of different packages for small businesses all the way to major corporations.

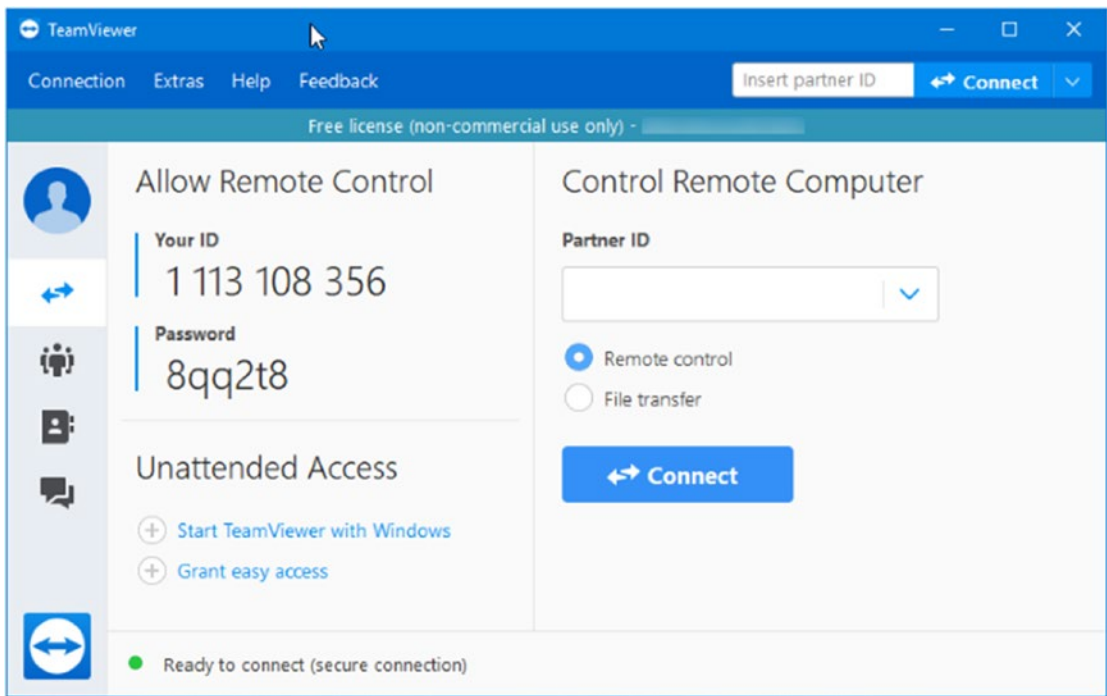


Figure 16-5. *TeamViewer is truly cross-platform*

TeamViewer is unlike the bundled Microsoft solutions too, as it offers a host of additional functionality from cross-platform access to reboot support, Virtual Private Network (VPN) support, ticket management, and support for deployment of applications across many computers and devices on a network.

RealVNC

Similar to TeamViewer, RealVNC (see Figure 16-6) is available for Windows (though only as a win32 installer), Mac OS X, Linux, iOS, Android, and Chrome OS and also adds support for Raspberry Pi, Solaris, HP-UX, and AIX into the mix. Again there are free and paid-for versions that fit any type of business and any type of budget. It can be found online at www.realvnc.com.

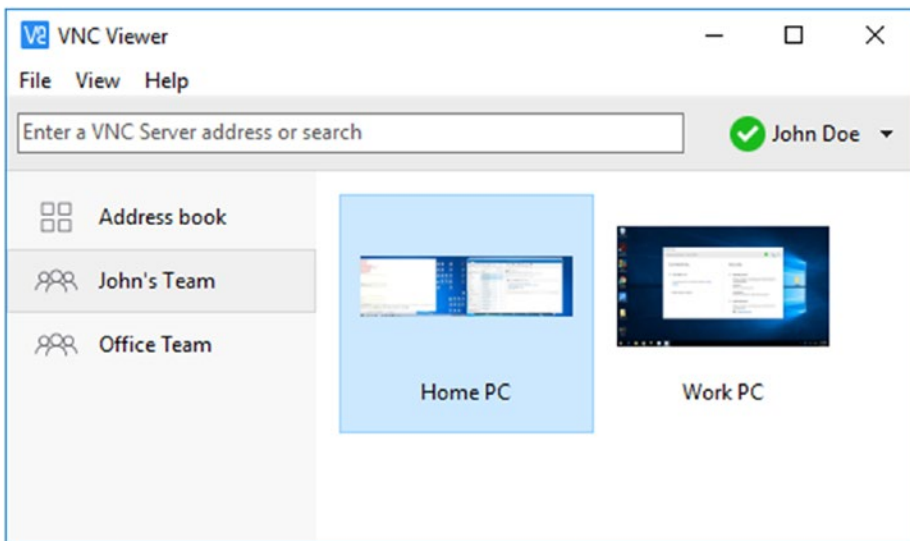


Figure 16-6. *RealVNC offers support for additional platforms*

RealVNC offers functionality similar to TeamViewer, with teach support, remote deployment, multi-lingual support, and reboot support.

LogMeIn

Another hugely popular remote access tool, LogMeIn (see Figure 16-7, available from www.logmein.com), is another cross-platform solution covering Windows (both as a win32 installer and as a Microsoft Store app), Mac OS X, Linux, Android, and iOS, and there are free and paid-for packages with various functionalities.

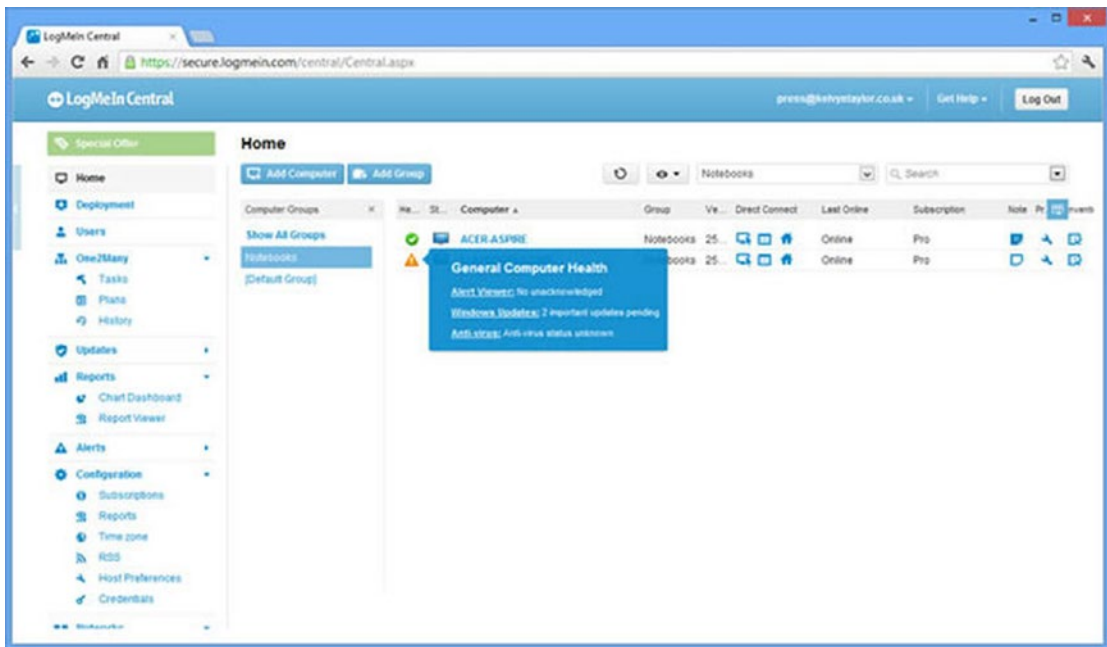


Figure 16-7. LogMeIn is another fully featured package

Again, and similar to TeamViewer and RealVNC, LogMeIn includes functionality not available in the bundled Microsoft packages, including cross-platform support and reboot support. The company also offers additional packages though that cover their other popular products, including the LastPass password security and management tool (both for consumers and businesses) and JoinMe online meeting service.

Chrome Remote Desktop

If you need to support people using Google Chromebooks, then the Chrome Remote Desktop plug-in for the Google Chrome web browser can be a good option. You can download Chrome Remote Desktop from <https://pcs.tv/2EJ4I0b> as a plug-in for the Chrome web browser. This means it's supported on Windows, Linux, Mac OS X, and Android and iOS installations that support browser plug-ins, as well as Google's own Chrome OS; see Figure 16-8.

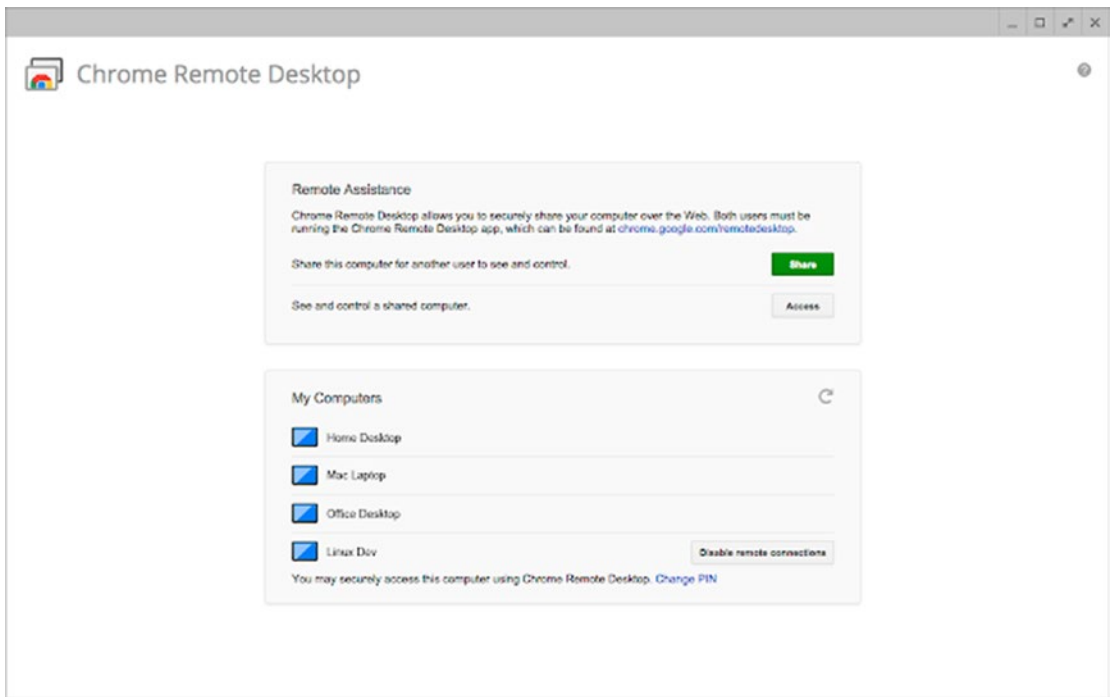


Figure 16-8. *Chrome Remote Desktop is a good solution for Chrome OS*

Chrome Remote Desktop supports cross-platform support, and it's completely free. This does mean though that it's limited in functionality when compared to its commercial rivals and even when compared to Windows Remote Assistance. There's no built-in chat, an inability to pass a Ctrl+Alt+Del command to the remote PC, no multi-session support, and no reboot support, and it requires both machines to be using the Chrome web browser.

Summary

This is by no means an exhaustive list of the tools available to you for remote support. Microsoft's own support department prefers using the BlueJeans Remote Desktop package, www.bluejeans.com/downloads, which supports Windows, Mac OS X, Linux, Android, and iOS and which comes in free and paid-for versions. There are always other options, and searching online might reveal a remote access solution that provides for your own needs more effectively than the packages I've detailed in this chapter.

Having the right tool for the job though is just one part of the puzzle, and so in the next chapter, I'll detail how you can go about using these tools to gather data and information about devices that you are supporting, remotely.

CHAPTER 17

Gathering Information Remotely

Earlier in this book, we looked at some of the tools available in Microsoft Windows to help obtain detailed technical and diagnostic information about a PC. What happens though if you're needing to obtain this information remotely?

Clearly, this presents its own set of unique challenges. You might be able to get full control of a PC with the user who is having a problem being already signed in using one of the tools I detailed in Chapter 16, but this isn't always going to be possible. I want to spend some time then examining some of the different ways that you can obtain technical and diagnostic information on IT systems.

Start with the Asset Tag

Most business and corporate IT equipment will come with an Asset Tag, though your business may call it something else. This is a code that will you have been assigned by the business to that specific piece of hardware that enables you to identify it.

This code will enable you to positively identify the hardware, assuming it was correctly catalogued to begin with, as it will give you the model number which you can use to search online for the full hardware specification. It won't however tell you about any changes or upgrades made to the equipment since, unless those changes and upgrades were also catalogued.

What it will tell you though, and again if the paperwork and reporting mechanisms that your business uses are effective, is the details of every IT support call and enquiry made about that device in the past. This information can be crucial to help determine the cause, and a quick solution to the problem, especially if it's a recurrence of a problem.

Additionally this will detail the names of the person or people who have reported these previous problems. I'm not suggesting you make aspersions about the people who use your equipment and systems, but a quick search might reveal that a particular individual tends to call IT support quite a lot. This could be because they have a tendency to break things, but it could every bit as easily be because they don't understand the systems very well. This can be mitigated to a certain degree by training, and how you respond to calls can feed back into your training regime, as I detailed earlier in this book.

If you are able to gain remote access to a PC, though perhaps the user experiencing problems isn't signed in, or if no user is signed into the PC, there are still things you can do to gain the information you need.

Permitting Remote Administration of PCs

In order to be able to work with PCs remotely, you first need to activate the Remote Administration functionality. This is disabled by default to enhance the security of systems, but it can easily be enabled in the Group Policy Editor.

To open the Group Policy Editor, search **gpedit.msc** in the Start Menu for **gpedit.msc**, then navigate within it to *Computer Configuration* ► *Administrative Templates* ► *Network* ► *Network Connections* ► *Firewall* and then, depending on how you connect to PCs in your business or organization, either *Domain Profile* or *Standard Profiles*. There you will see an option to **Allow [an] inbound remote administration exception**; see Figure 17-1.

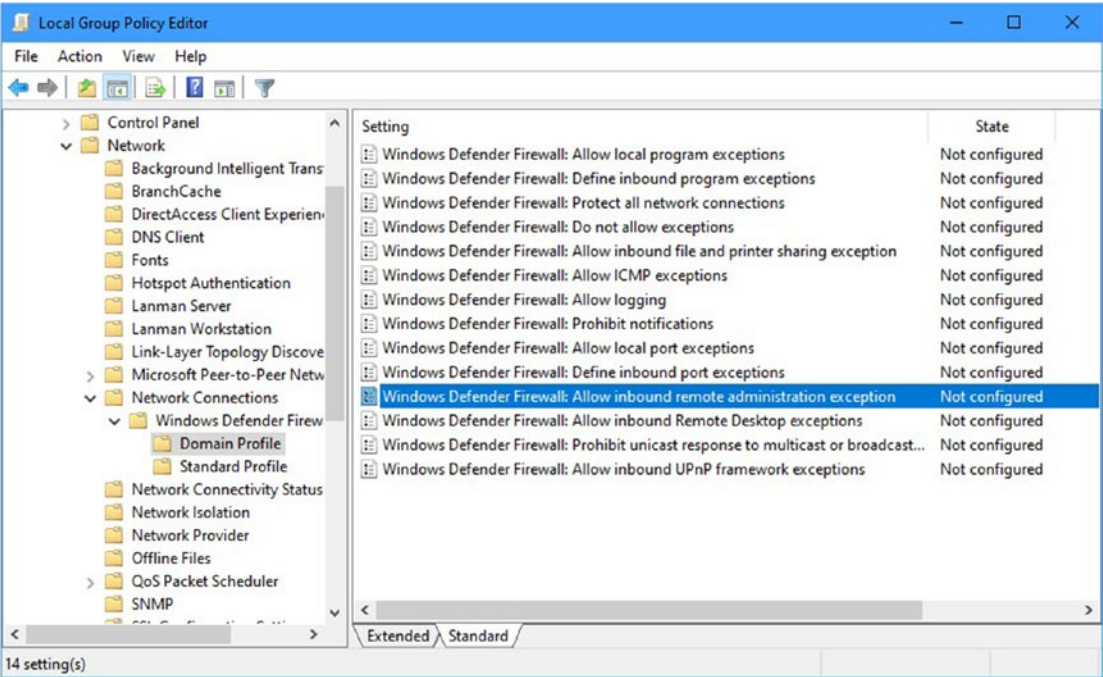


Figure 17-1. *You need to allow Remote Administration in Group Policy*

You should enable this policy, but it doesn't have to make the PC vulnerable, as you can specify which IP addresses Administration Access will be granted to; see Figure 17-2.

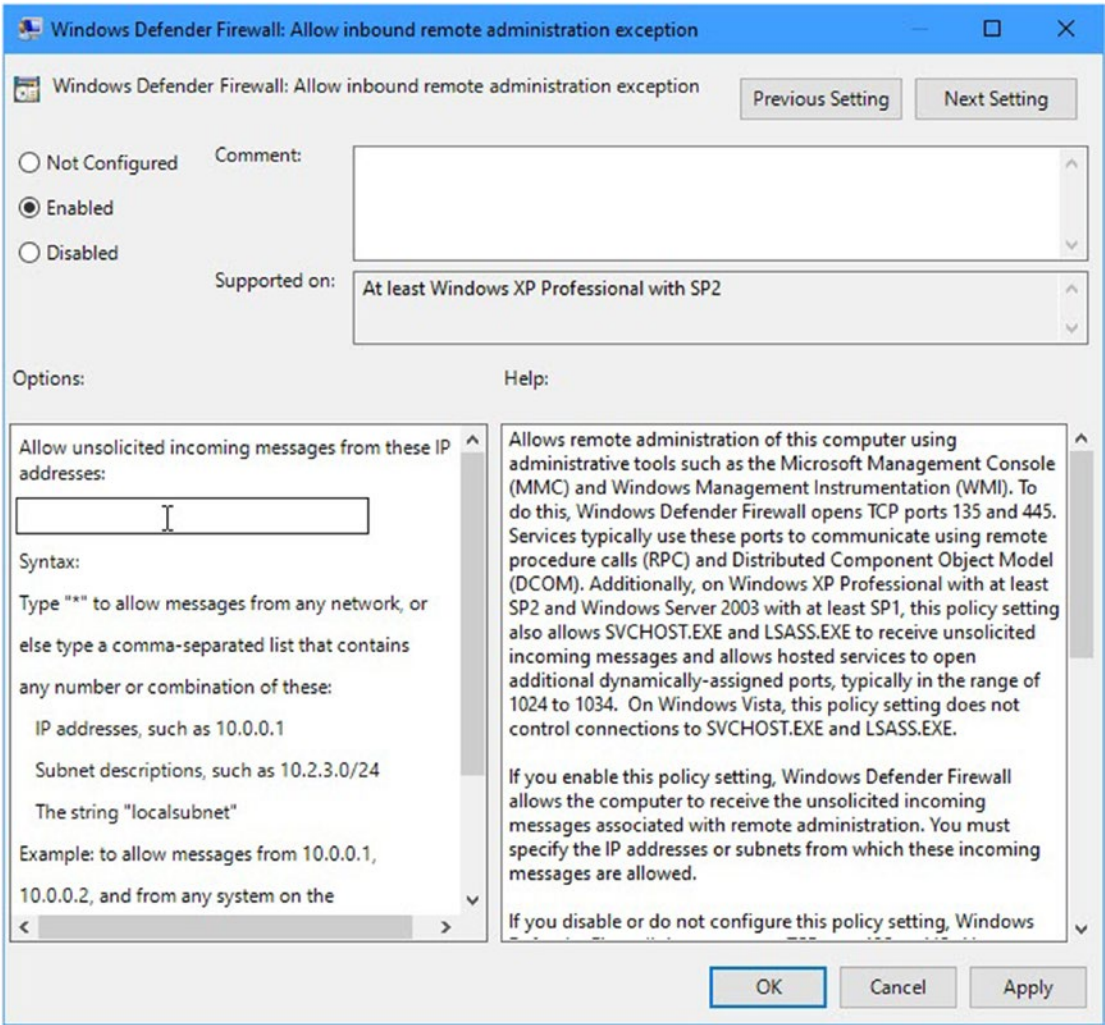


Figure 17-2. You can specify IP addresses for Remote Administration

Additionally, you will need to open two TCP ports for remote access. You do this in the *Advanced Firewall*; search for this in the Start Menu. With the Advanced Firewall open, click *Inbound rules* in the left-side panel, and then click the *New Rule* option in the right-side panel. Select *Port* in the dialog that appears and create a firewall exception for ports 135 and 445; see Figure 17-3.

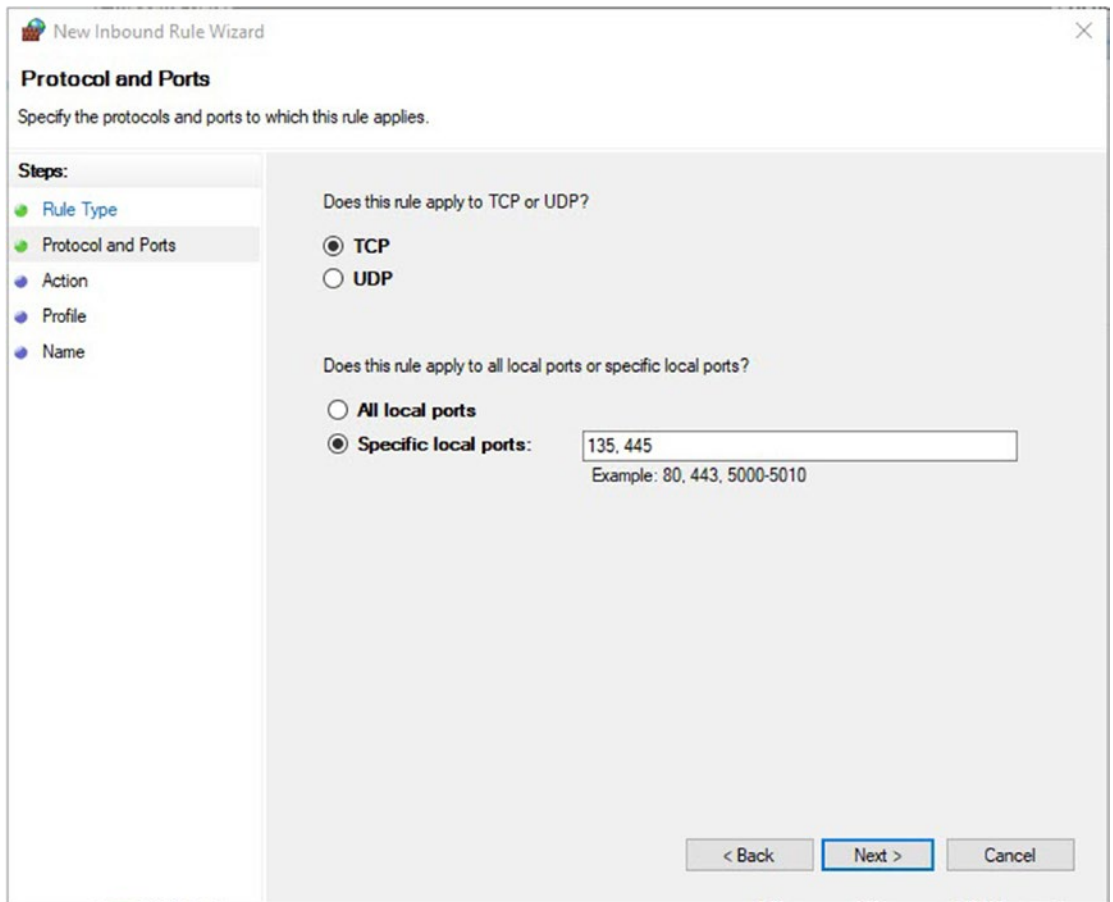


Figure 17-3. You need to open TCP ports 135 and 445

Sign in to the Registry as Another User

If you need to examine, or make changes to, the Registry for a specific user, you can sign in to the Registry remotely using their identification. Remember that a PC has its global Registry files, containing settings that affect the PC as a whole, but also each individual user has their own Registry files too, which contain all the settings for their own user account.

To remotely access the Registry for a user on a PC, that PC needs to be running the *Remote Registry* service. You can do this from the *Services* panel (search for services in the Start Menu) and by enabling *Remote Registry*; see Figure 17-4.

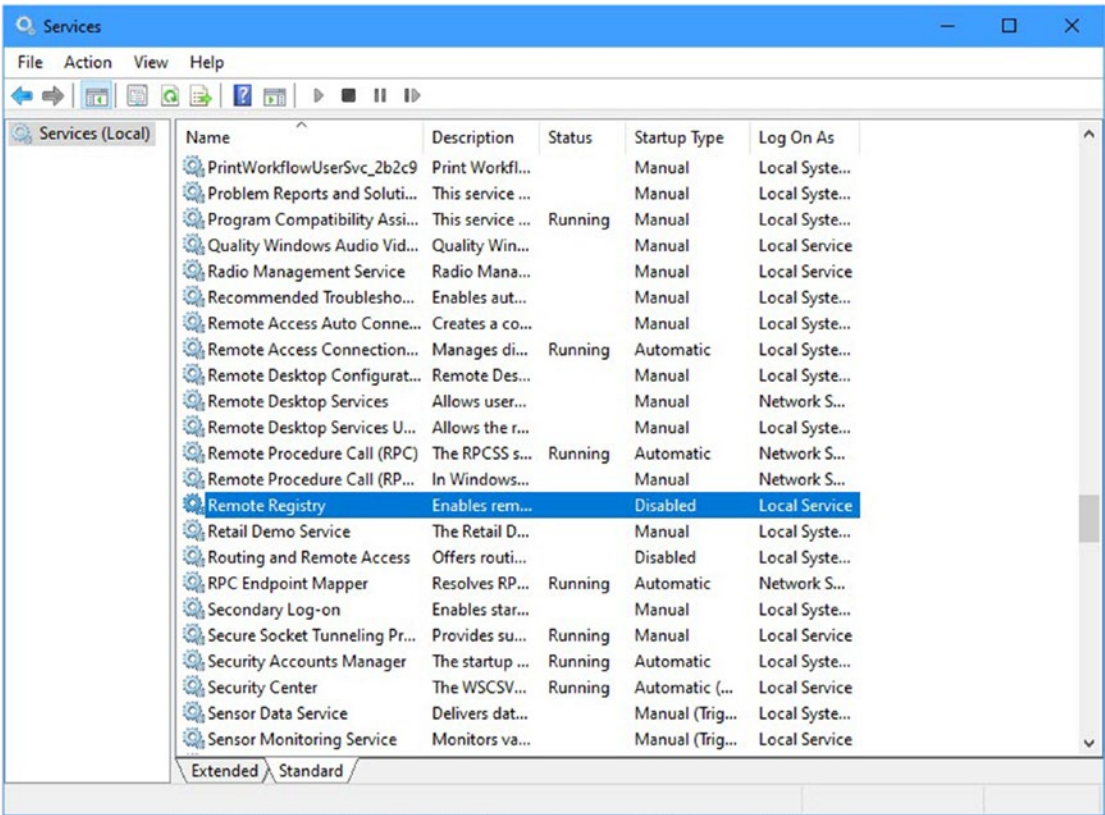


Figure 17-4. You can access Remote Registry in Services

Alternatively, you can start the Remote Registry service from an Administrator Command Prompt, using the command **sc start RemoteRegistry start = auto**.

With these services and rules in place, you can connect across your network from the Registry Editor (Regedit at the Start Menu) to the remote PC. From the File menu in the Registry Editor, select *Connect Network Registry* (see Figure 17-5), and enter the details of the remote PC you wish to connect to and the user whose Registry you need to access.

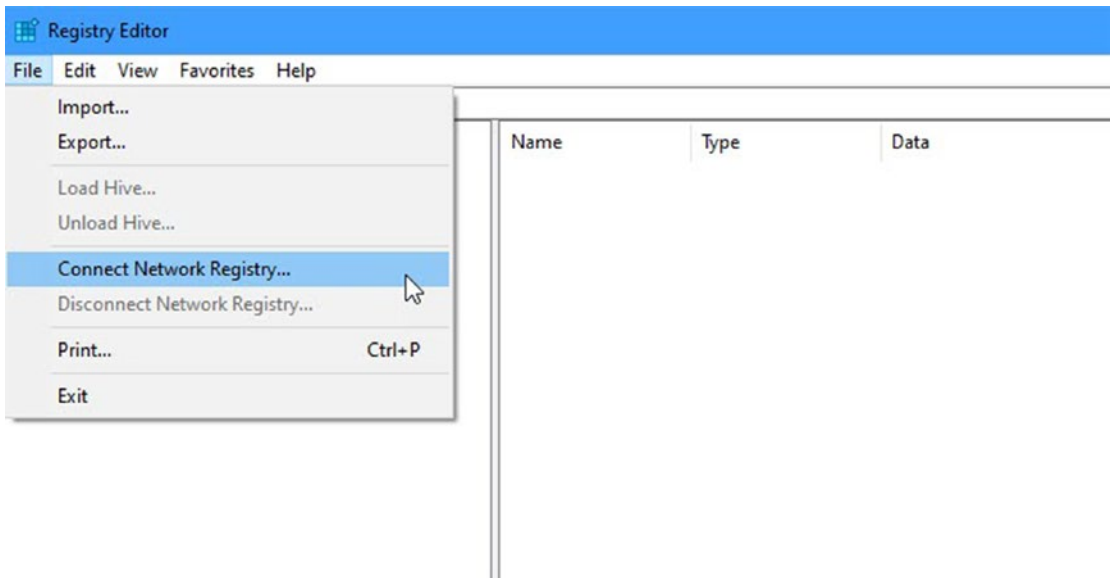


Figure 17-5. You can connect to Remote Registries

Using the Microsoft Management Console Remotely

The Registry isn't the only part of the Windows operating system that you can access remotely. With Remote Administration enabled in Group Policy, you can also access the full Management Console for a remote PC. Open *Computer Management* on your own PC, and from the *Action* menu, select *Connect to another computer*; see Figure 17-6.

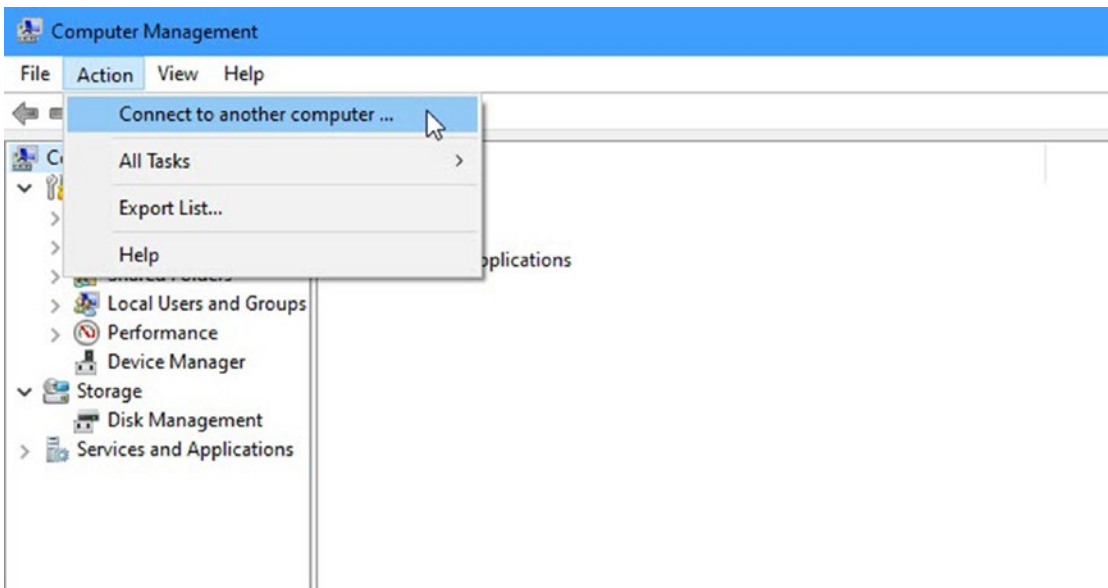


Figure 17-6. You can connect to the Management Console of other PCs

From this Management Console, you have full access to Administrative tools including the Event Viewer, Task Scheduler, Device Manager, Performance Monitor, Services panel, and Disk Management tools. These tools can provide you with a huge amount of information, but should you require even more, then the *System Information* panel, available by searching for it in the Start Menu, can also be accessed across a network (see Figure 17-7) by choosing *Remote Computer* from the *File* menu.

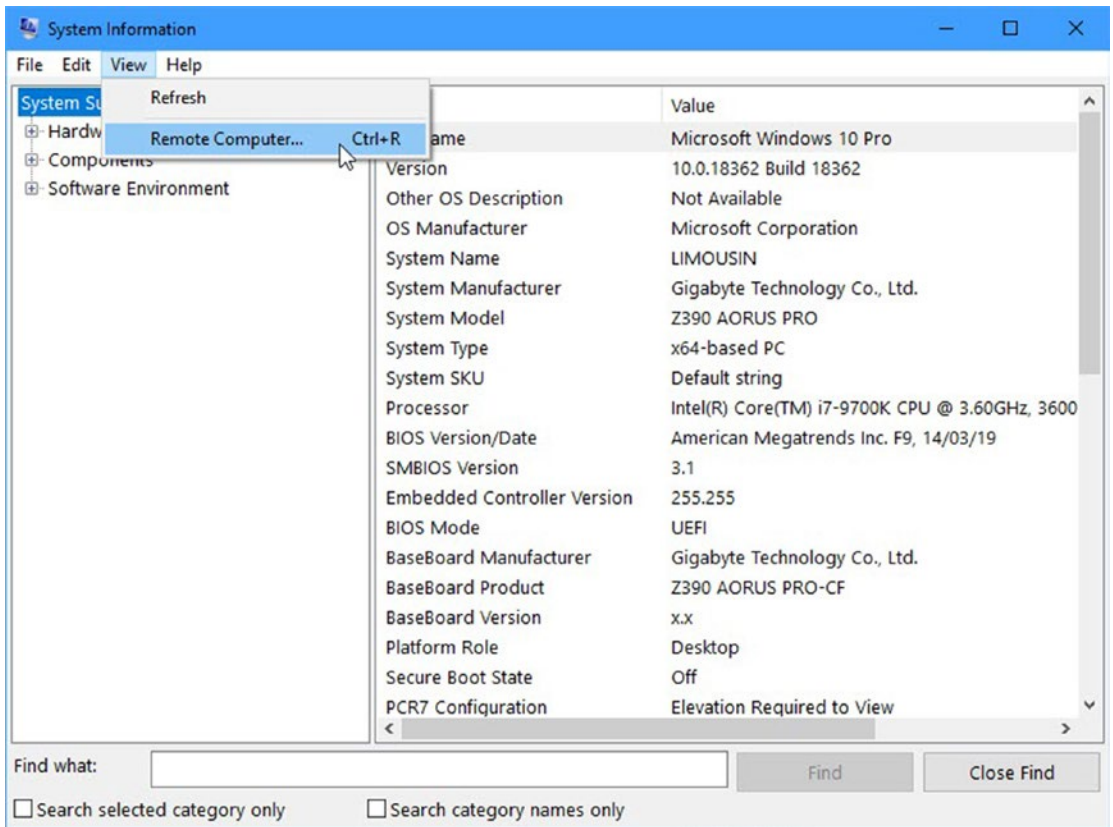


Figure 17-7. You can access System Information remotely

Summary

Being able to sign in to a PC remotely, even as a specific user, to gain information about the system is incredibly useful. This is partly due to the convenience it offers you but also because the end user really doesn't want to be sitting there the whole time watching their cursor move around the screen on its own.

It's true then that Remote Administration opens up many useful possibilities for you to be able to get problems fixed quickly and efficiently. What if you find yourself in a position though where you simply cannot gain remote access to a PC and where you're entirely at the mercy of the end user to help you? This is what we'll look at in the next chapter.

CHAPTER 18

Helping Your Users to Help You

Providing IT support can be a hugely complex business; it's not something to be undertaken lightly, as it can be very involved and hugely technical and require both focus and skill. This means then that having the end user help you with the process, and perhaps even fixing problems themselves, would be the last resort.

Sadly, because you may not be able to gain remote access, or use other methods of diagnosing and troubleshooting a problem remotely, or an engineer might not be available, the last resort may also have to be the first response.

So bearing in mind that throughout this book I've talked about how to speak to users in non-technical ways, how to get them to help you understand what the problem is, and how you can explain to them how they can avoid the problem from recurring, or even sometimes fix it should that happen; how do you get them to provide the support for you if there's simply no other choice?

Problem Steps Recorder

The Problem Steps Recorder, found by searching in the Start Menu for **psr**, is one of the hidden gems in the Windows operating system. Originally designed as a tool for use by beta testers for Windows 7, it proved so popular that Microsoft kept it in the final release of the OS and continued supporting it in Windows 8.1 and Windows 10.

The Problem Steps Recorder consists of a simple-to-use toolbar (see Figure 18-1) that allows end user to record everything that happens on their computer screen, and it then annotates this with more technical information from the backend.

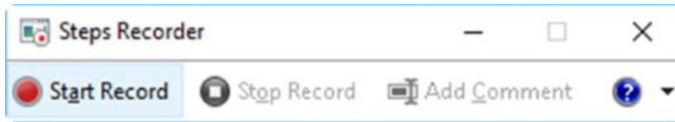


Figure 18-1. *The Problem Steps Recorder*

The end user just has to press the *Start Record* button (see Figure 18-2) and then do whatever it was they were doing when the error or problem occurred, so that they can replicate it. This isn't recorded as a video, so there's no need for them to rush through things.

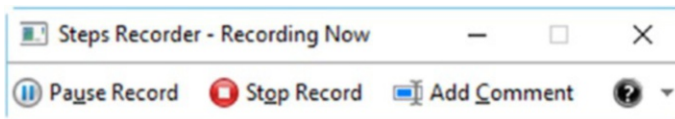


Figure 18-2. *The PSR controls are easy to use*

When they press the *Stop Record* button, the recording will be saved as an MHTML file, contained within a standard compressed archive (ZIP) file, which they can email to you. It contains screenshots and text descriptions of everything that occurred; see Figure 18-3.

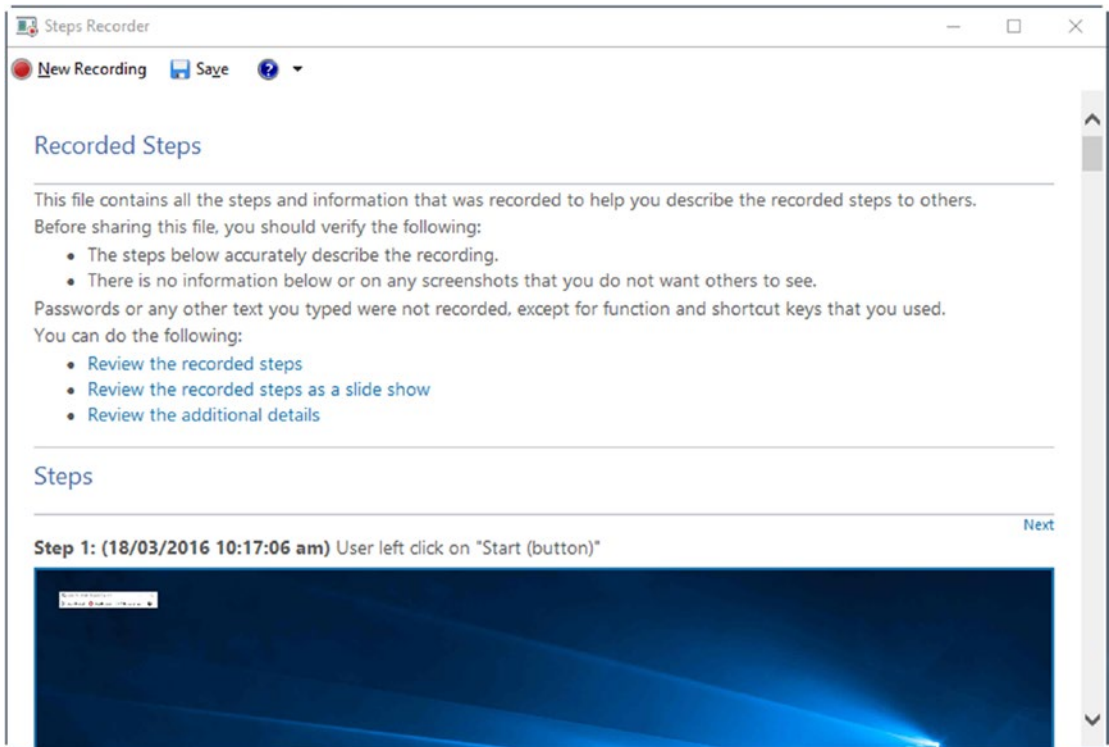


Figure 18-3. PSR reports are easy to follow

In each screenshot, whatever it is that is clicked, or that has changed, is annotated; in Figure 18-4 you can see a green line has been drawn around the Start Menu, which indicates that the action being recorded is that the menu was opened by the user.

Step 2: (18/03/2016 10:17:09 am) User left click on "All apps with new apps available (button)" in "Start"



Figure 18-4. *Everything that is clicked or changes on screen is annotated*

From the main toolbar, the user can also click the *Add Comment* button. This freezes everything on their screen and allows them to type a comment or notes for you if they believe something pertinent has happened or if they have a question to ask (see Figure 18-5).

[Previous](#)

Step 12: (18/03/2016 10:17:56 am) User Comment: "What's this Direct 3D thing and how do I get it?"

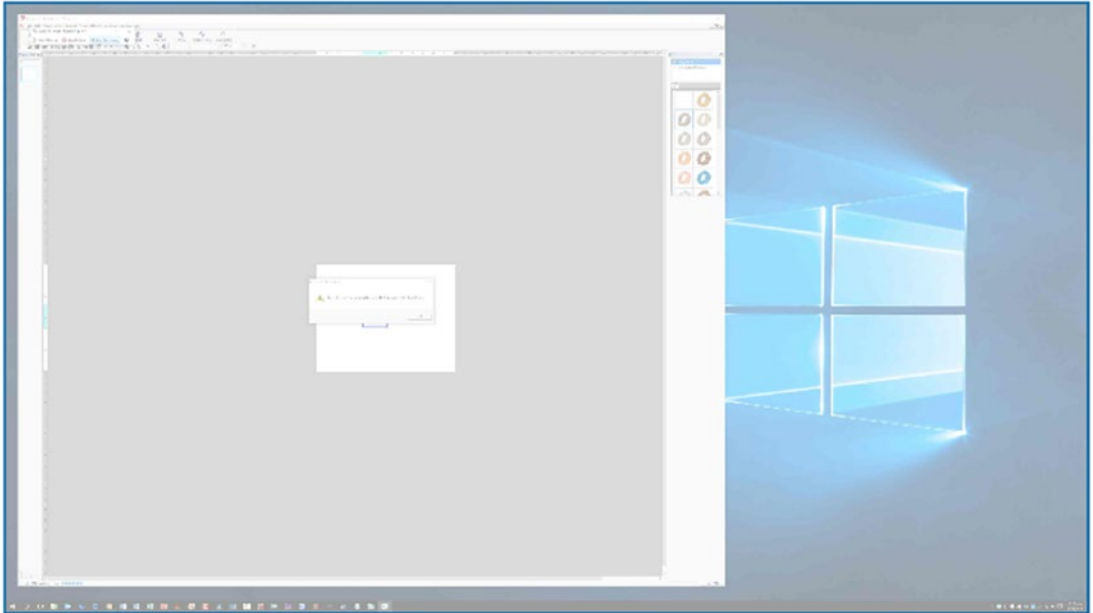


Figure 18-5. *You can place comments throughout the recording*

At the very bottom of the recorded actions log, the Problem Steps Recorder also includes more technical information about the actions occurring on screen and the status of both the PC and any running apps and services at the time; see Figure 18-6. This can be especially useful for drilling down into specific actions and events on a PC.

Additional Details

The following section contains the additional details that were recorded. These details help accurately identify the programs and UI you used in this recording. This section may contain text that is internal to programs that may only be understood by very advanced users or programmers. Please review these details to ensure that they do not contain any information that you would not like others to see.

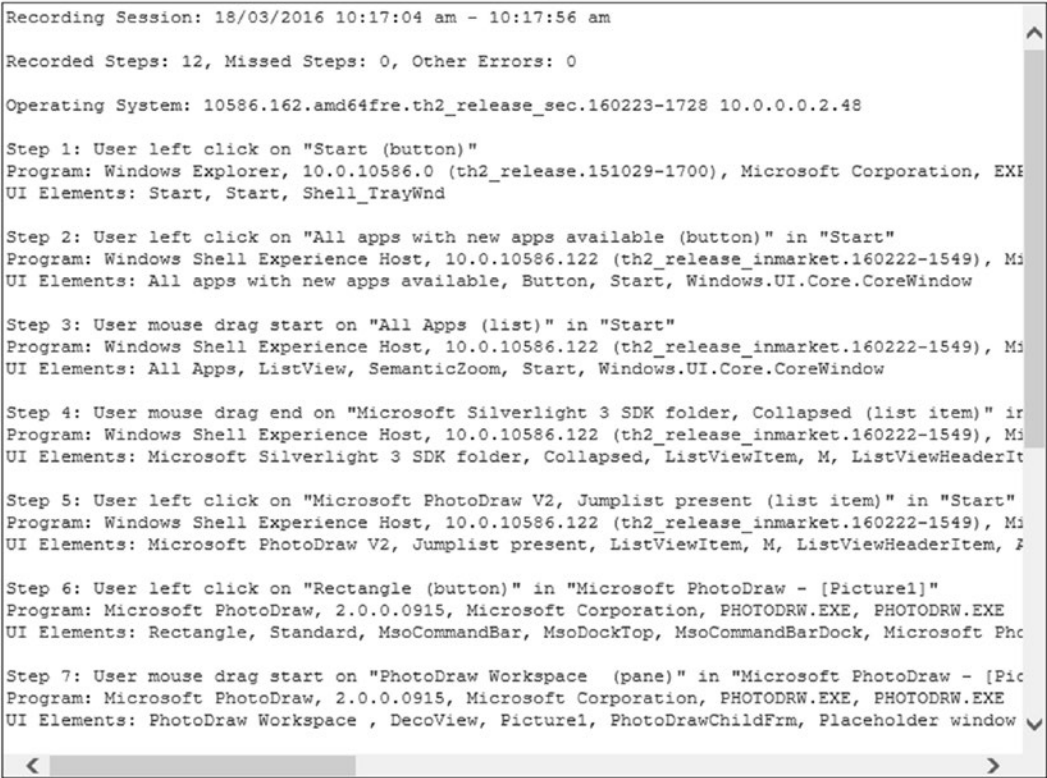


Figure 18-6. Text descriptions of everything that happened are also recorded

As I mentioned a little while ago, the user can email the provided ZIP file, which only contains images and text, to you or to another support person, and the beauty is that the Problem Steps Recorder can be found in all supported versions of Windows.

Saving Screenshots

A great many people will be aware that the **PrtScrn** (Print Screen) button on keyboards can be used to capture a screenshot on PCs, but this method saves the screenshot to the computer's clipboard, and it then has to be pasted into a document and saved.

Windows 8.1 and 10 include an addition feature whereby you can press **Win + PrtScrn** (the Windows key and Print Screen) and the screenshot will then be automatically saved to a *Screenshots* folder within the user's *Pictures*; see Figure 18-7. Additionally, on a Windows tablet you can press the Windows button and volume down to achieve the same thing.

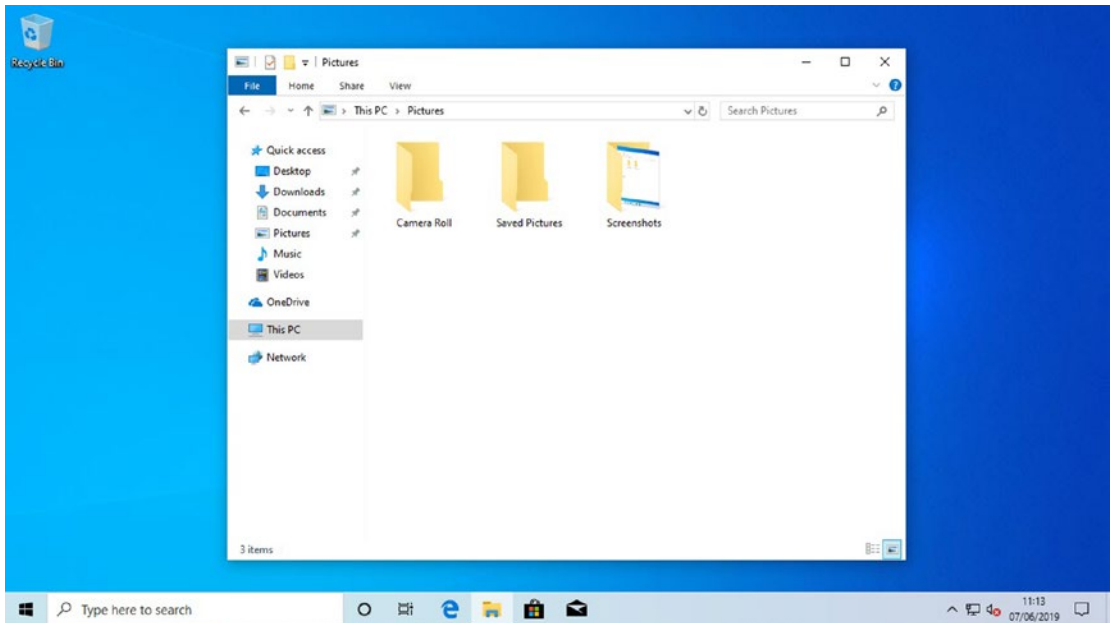


Figure 18-7. *Win + PrtScrn automatically saves screenshots*

The process is different on other operating systems:

- **Chrome OS** – Press Ctrl plus the Windows Switcher button.
- **Mac OS X** – Press Shift-Command and 5.
- **iOS** – Press the side button and the volume up button.

- **Android** – Press the power and volume down buttons, though some Android devices also have a dedicated screenshot ability.
- **Linux** – Press PrtScrn and you will be asked if you want to save the screenshot file.

Screencasting

Something else of note that can prove useful for operating systems other than Windows is screencasting and screen recording software. These are available for all operating systems and include everything from the professional Camtasia from www.techsmith.com to cheap and even free apps available from the Apple and Google stores.

If you need a video showing you what has happened on a non-Windows device, these apps can be valuable, and they are frequently usable by non-technical people. I won't make recommendation about specific Android, Chrome OS, and iOS apps as their functionality and specification can change rapidly, but it can be worth finding apps you can then recommend to your end users.

Xbox Game Bar

For recording video on PCs, I'll bet you didn't think I'd wrap this up by talking about a gaming tool. On Windows 10 you can access the **Xbox Game Bar** from the Start Menu, unless it has been blocked by your corporate policies.

The Xbox Game Bar is a very easy to use tool that is similar in look and functionality to the Problem Steps Recorder; see Figure 18-8. However it records video of what happens on your PC screen.

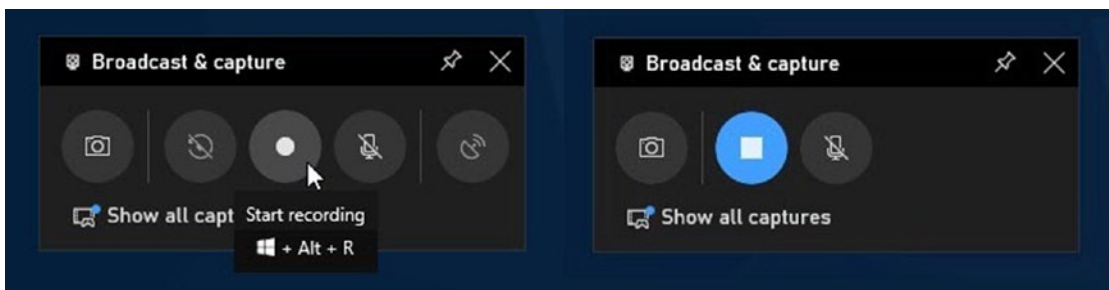


Figure 18-8. The Xbox Game Bar records video from your PC

The resulting video file is then saved in the *Videos ► Captures* folder and can be either emailed to yourself or another support person, uploaded to cloud storage or a server, or transferred on a USB flash drive.

Note though that there are limitations with the Xbox Game Bar, the biggest being that it will only record a *single* application. This makes it great if you want to see what the problem is with a certain app, such as a custom accounts package, but it's useless if you need to get a recording of the whole PC desktop and Windows interface.

Summary

Armed with this information, and these tools, you will now be able to have your users provide you with quality information no matter what the circumstances, even if it's them taking a photo of a hardware problem with their own smartphone and sending it to you on a messaging app such as Microsoft Teams.

You now have all the information you need to be able to approach any technical or troubleshooting problem, be it with hardware, software, and on any type of operating system, anywhere in the world, and with any type of end user, with professionalism and aptitude.

Remember though that this book is only your starting point. You will need to take what you have learned here and adapt it to suit your own organization, processes, and the specific hardware and software you support. This will also be a constantly evolving process, but if you allow that evolution to happen naturally and regularly, you will find that the training and support that you are able to offer people will always be first class and will mean that you will always remain at the top of your game ... good luck.

Index

A

- Accessibility, [79](#)
- Administrative tools
 - data collector sets, [141–143](#)
 - definition, [136](#)
 - event viewer (*see* Event viewer)
 - performance monitor, [138–141](#)
 - system information, [136–138](#)
 - utilities, [136](#)
- Advanced Research Projects Agency
 - Network (ARPANET), [16](#)
- Android, creation, [70](#)
- Asset Tag
 - definition, [173](#)
 - Registry remotely
 - administrator command, [178](#)
 - Registry Editor, [178, 179](#)
 - Remote Registry service, [177, 178](#)
 - remote access, [174](#)
 - remote administration (*see* Remote administration of PCs)
 - reporting mechanisms, [173](#)

B

- Beta testers, [183](#)
- BlueScreenView, [159](#)
- Built environment
 - bluetooth/cellular signals, [100, 101](#)
 - Wi-Fi connections, [99, 100](#)

C

- Cathode ray tube (CRT), [14](#)
- Chrome remote desktop, [170, 171](#)
- Computers
 - history, [13, 63, 64](#)
 - ARPRANET, [16](#)
 - CRT, [14](#)
 - electronic device, [13](#)
 - Gemini PDA, [16](#)
 - Olivetti M240 PC, [15](#)

D

- Data collector sets, [141–143](#)
- Distributed denial of service
 - attacks (DDoS), [10, 72](#)
- Documentation, training
 - education/technical understanding, [108](#)
 - language, [109](#)
 - materials, [107, 108](#)
- Documentation, troubleshooting, [109](#)
 - engineering solutions, [111](#)
 - paperwork, [111](#)
 - SLA, [110](#)
- Dynamic Host Control Protocol (DHCP), [65](#)

E

- Engineer paperwork
 - faulty network socket, [130](#)

INDEX

Engineer paperwork (*cont.*)

- feed back, [130](#)

- IT support personnel, [129](#)

Environmental factors

- ingress protection (liquids) ratings, [98](#)

- ingress protection (solids) ratings, [97](#)

- weather, [95](#), [96](#)

Error and status information

- contents of event logs, [155](#), [156](#)

- custom view, [152](#), [153](#)

- event information, [154](#)

- event property, [155](#)

- event viewer, [151](#), [152](#)

- timeframes, [152](#), [153](#)

Events tab, [91](#)

Event Trace Log (ETL), [158](#)

Event viewer, [57](#)

- administrative events, [144](#)

- attaching tasks to

 - events, [147](#), [148](#)

- critical events, [144](#)

- custom views, [145](#), [146](#)

- definition, [143](#)

- usage, [148](#)

- windows logs section, [145](#)

Extensible Markup Language (XML), [158](#)

F

First-line support paperwork

- checklists, [125](#)

- diagnose common problems, [123](#)

- hardware/software,

 - specification, [124](#)

- valuable information, [126](#)

Flow logic/troubleshooting

- asset tag, [41](#)

- diagnostic process, [42](#)

- eliminate problem, [37](#)

- information, [40](#)

- paperwork, [41](#)

- problem, cause, [42](#)

- process of elimination, [38](#), [39](#)

- team work, [43](#)

G

Graphical user interface (GUI), [67](#)

H

Hardware, [73](#), [74](#)

Human

- achievement, [82](#)

- training, [82](#)

Human factor, [11](#), [12](#)

I, J

Interconnectedness, IT systems, [9](#), [10](#), [20–22](#)

Interface standards, [17](#)

- firewire, [18](#)

- operating systems, [20](#)

- parallel devices, [19](#)

- serial devices, [18](#), [19](#)

- types, [19](#)

- USB, [17](#), [18](#)

Internet

- interconnectedness, [71](#)

- social media, [71](#)

- troubleshooting, [72](#)

- working, [73](#)

IPv4 networking system

- DHCP, [65](#)

- home router, [65](#)

IT issue
 diagnostic process, 55, 57
 example, 58, 59
 information, 57
 networking, 54
 troubleshooting, 56
 USB peripheral, 54
 working backward, 55

IT support
 billing, 121
 fundamentals, 5
 assumptions, 8
 basic principle, 5
 effects/ramifications, 7
 language barrier, 8, 9
 problem, 7
 tracing problems, 7
 PC problems, 4
 protocols, 4
 repository, 121
 tracking, 121

IT systems
 hardware, 76–78
 settings, 79
 software, 78

IT troubleshooting problem, 59

K

Knock-on effect, 71

L

Learning theory
 alert notification, 29
 assessment, 27
 esteem needs, 28
 IT support, 26
 kinesthetic, 27

Maslow's hierarchy of needs, 28
 network connection
 problem, 29
 security policy, 29
 self-actualization, 28
 testing, 27
 written/online evaluation, 26

Legacy devices

 adding to Windows, 84, 85
 configuration, 87, 88
 driver, selection, 87
 types, 86
 analogue, 84
 peripherals, 83

LogMeIn, 169, 170

M

Microsoft DOS, 67
 Microsoft Management, remote
 access, 179–181
 Microsoft SysInternals suite, 58

N

Network attached storage (NAS), 10
 Non-technical dictionary, 51
 Non-technical individuals, 80

O

Online chats, 51, 52

P

Performance monitor, 138–141
 Peripherals, 92
 Personal digital assistant (PDAs), 10
 Personal security certification, 66

INDEX

Problem Steps Recorder
comment button, [186, 187](#)
reports, [185](#)
screenshot, [185, 186](#)
search psr, [183](#)
Start Record button, [184](#)
Stop Record button, [184](#)
technical information, [187, 188](#)
toolbar, [183, 184](#)
ZIP file, [188](#)

Q

Querying users
access websites, [48](#)
initial assessments, [47](#)
IT support, [45](#)
language barrier, [45](#)
regional dialects, [45, 46](#)
technical questions, [48](#)
users, kind of, [46, 48–50](#)
Quick assist, [166, 167](#)

R

RealVNC, [169](#)
Reliability
error reports, [135, 136](#)
history, [133](#)
information events, [134](#)
technical information, [135](#)
tools, [134](#)
Reliability monitor, [57](#)
Remote administration of PCs
advanced firewall, [176](#)
gpedit.msc, [174](#)
Group Policy Editor, [174, 175](#)
IP addresses access, [175, 176](#)
TCP port, [176, 177](#)

Remote computer option, [138](#)
Remote desktop, [163–165](#)
Remote support tools
chrome remote desktop, [170, 171](#)
LogMeIn, [169, 170](#)
quick assist, [166, 167](#)
RealVNC, [169](#)
remote desktop, [163–165](#)
TeamViewer, [168](#)
windows remote assistance, [165, 166](#)
Resource monitor, [58](#)

S

Screencasting, [190](#)
Screenshots
operating systems, [189, 190](#)
PrtScrn, [189](#)
Win + PrtScrn, [189](#)
sc start RemoteRegistry start = auto
command, [178](#)
Second-/third-line support paperwork
engineers, [128, 129](#)
troubleshooting, [127](#)
upper-level support technicians, [127](#)
Security vigilance, [50](#)
Self-contained module, [30](#)
Special Executive for Counterintelligence,
Terrorism, Revenge and Extortion
(S.P.E.C.T.R.E.), [39](#)
Staff training, [26](#)

T

Task manager
high configuration, [149](#)
performance tab, [148](#)
processes tab, [148](#)
TeamViewer, [168](#)

Training course

- assess learners, [31](#)
- Écouter et Répéter, [33](#)
- evaluation, [33](#)
- learning styles, [32](#)
- mixed peer groups, [31](#)
- objectives, [30](#)
- performing activities, [30](#)

Troubleshooting

- disadvantages, [116](#)
- flowchart, [116](#)
- flow logic, [117–119](#)
- SLAs, [116](#)
- step-by-step approach, [119](#)
- structured teams, [115](#)

Troubleshooting device drivers

- different types, [89](#)
- error reporting information, [89](#)
- errors/events, [92](#)
- information, [91](#)
- roll back, [89](#)
- system restore, [88](#)

U

Uninterruptable power supply (UPS), [74, 96](#)

Universal serial bus (USB), [67](#)

Universal Windows app

- Platform (UWP), [69](#)

Upshot, [71](#)

User Account Control (UAC), [69, 166](#)

Users

- communication, [24, 25](#)
- human factors, [24](#)

V

Virtual machine (VM), [10](#)

W

Windows log files

- Dmp files, [158, 159](#)
- text log files, [157](#)
- XML and ETL, [158](#)

Windows NT, [67](#)

- advantages, [67](#)
- UAC, [69](#)
- USB, [67](#)

Windows on ARM (WoA), [69](#)

Windows remote assistance, [165, 166](#)

Windows vNext (version Next)

- problem, [70](#)
- WoA, [69](#)

X, Y, Z

Xbox Game bar, [190, 191](#)