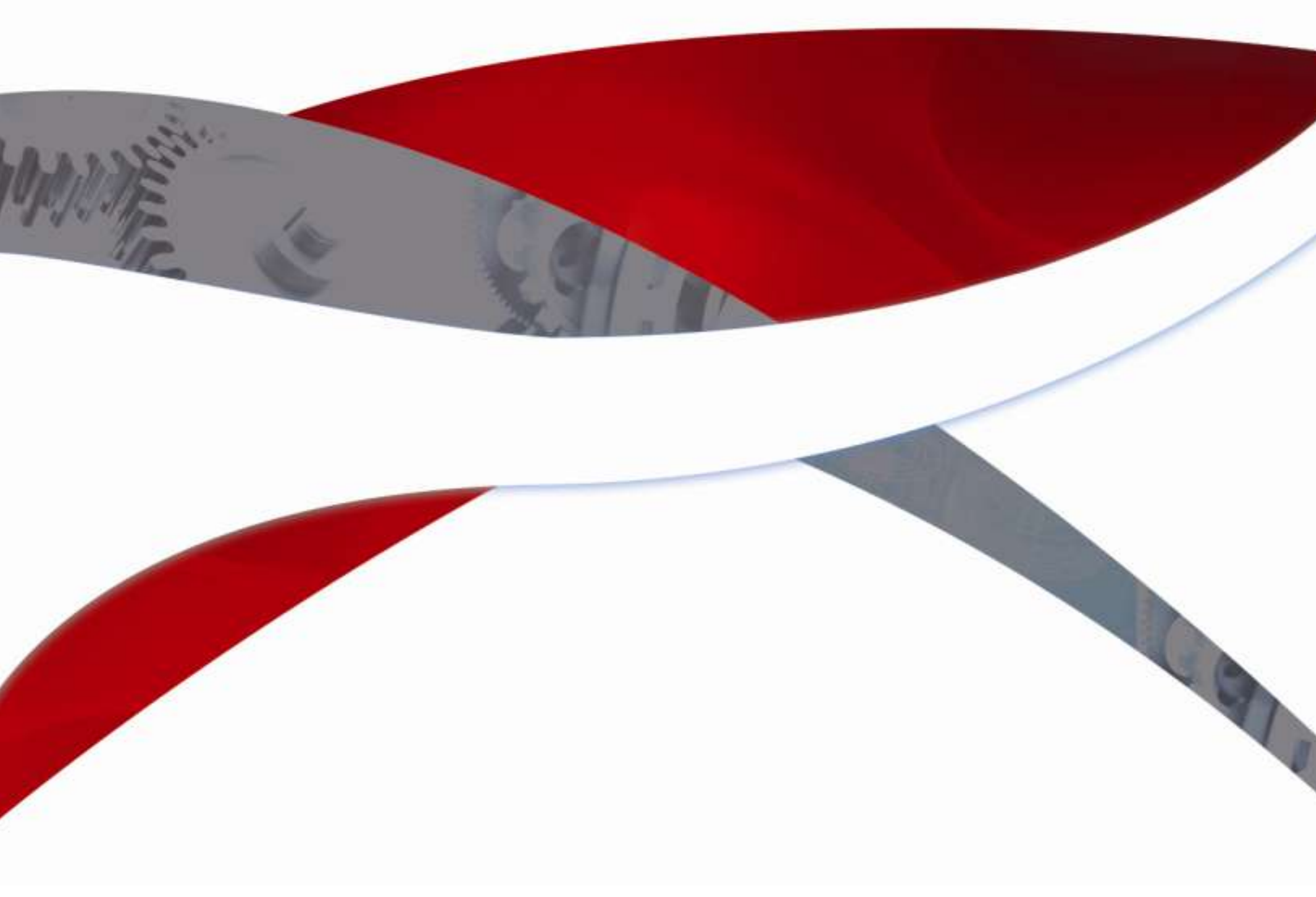
White Paper

The PC Performance Experts™



The Battle for Workstation Productivity

And the Tools You Need Win It

The Battle for Workstation Productivity and the Tools You Need to Win It

In this white paper iolo technologies discusses the critical role of Windows workstations in the never-ending battle to achieve higher levels of business productivity. The report delves into the issues faced by business owners, managers, and IT professionals as they work to keep these systems up and running. There is no longer any question that desktop technology is an essential part of doing business anywhere in the world today; no one argues this point. But when it comes to determining the most effective ways to set up, manage, and maintain these systems, iolo has observed that there are still some very strong differences of opinion. While management styles may vary, the underlying concerns about the time, money, and other resources expended on workstation maintenance are similar across all types and sizes of companies in a broad range of industries.

The tools for workstation performance optimization iolo has been developing since 1998 are based in large part upon many of these common-sense concerns. The technical or machine-oriented capabilities of the iolo solution are complemented by their ergonomic, people-oriented qualities, adapting to the needs of users and managers in small, medium, and large enterprises. Business executives seeking to evaluate and implement workstation-optimization tools will benefit from a review of the issues faced by so many other companies in similar situations. In this paper iolo technologies also addresses several commonly held misconceptions about workstation performance degradation and what, if anything, can be done to prevent it. The reader will in the end be able to see how each of these factors has influenced the development of the System Mechanic® suite of tools, the ultimate goals of which are to a) streamline the configuration of the workstation to obtain optimum performance, b) prevent unnecessary downtime, c) extend the useful life of the machine, and d) forestall the expense of total replacement.

Intelligent Productivity and the Management Mindset

In 1998 iolo technologies began developing a series of software tools aimed at correcting numerous flaws in the configuration of Windows workstations. The goal was to improve speed and performance, prevent downtime, and extend the useful life of the computer. In the course of this endeavor iolo has engaged extensively with users and managers in every type of environment, seeing first hand how their systems react to different styles of working. One of the most consistent observations made in these engagements is that high productivity alone does not make a business successful, and the same is true of software solutions. Manufacturing at higher and higher rates, or executing more and more lines of computer code per second, does not necessarily produce better results. Higher productivity, to be effective, must be guided by better intelligence. A common example is when a highly productive business is not tightly focused on the right products for their target markets; they are able to produce more than they can sell, an out-of-balance condition. The battle for productivity must include the equally important battles for product usefulness, and market relevance, along with technical accuracy, cost-efficiency, and quality control.

The decision-makers who evaluate workstation productivity solutions and other business investments tend to operate on a similar mindset. Whether buying land, buildings, machinery, materials, hiring employees, or selecting business partners, the factors they consider are always the same: Is this person, product, or activity relevant to my line of business? If so, what will it cost to own and operate, and how will it benefit the company? In the end what will be the bottom-line result, the financial rate of return? The negatively constructed version of this line of thinking is in mind at the same time: Can I get along without this person or technology and what will happen, really, if I simply put the matter off until later, or forget it entirely? This is not to say that a company never buys something it doesn't really need, simply because they wanted it, such as a premium Italian espresso maker; they sometimes will, but this is the exception and not the rule. In cost-conscious companies, and in any firm during tight financial times, the vast majority of expenditures must be accompanied by strong financial arguments.

And then there are the technical issues, which very often come before the financial review. What will the solution do? How does it work? Can I see inside it? What level of control will I have? Is it compatible with our hardware and software environments and our types of users? How easy is it to install and configure? Can it be rolled back? How long has it been operating in the field in similar environments and what has been its track record? You don't have to be an IT professional in a 1000-seat enterprise to ask these questions. In iolo's experience, the owners and managers of small and medium-sized companies usually express a very similar list of concerns.

Protecting the Business Investment

A business runs like a speedboat on the twin engines of purpose and productivity. Without a clear, consistent purpose, there is no basis for ramping up into higher and higher levels of productivity, and without a clear path to productivity; the company's core purpose cannot be achieved. With the purpose set and the investments made, it now becomes increasingly important to take the actions needed to protect those assets. In iolo’s experience, however, this is where the decision-making process in many companies becomes fuzzy. The reason seems related to the fact that solutions designed to protect an investment are “less visible” and therefore more difficult to assess than the original equipment itself. Anyone can look directly at a car or a trained elephant or a new computer and see clearly, by hands-on demonstration if necessary, what it will do and why it is valuable. But without direct experience over the life of the equipment, it is not nearly as easy to envision the value of a product warranty or a maintenance service plan. People need cars and want cars, but when it comes to performing routine maintenance, they need it but they don’t really want it. To make the case for workstation maintenance and tune-up, there must be a clearly visible link to business productivity and profits. It has to be a fast and easy solution or it will be perceived as unworkable. Consider the position of Microsoft Corporation on the subject, in an article about Windows XP workstations:

> *“Computers, like cars, homes, and appliances, need regular maintenance to keep functioning well. Fortunately, keeping your computer well maintained isn't a time-consuming task because most of the work can be completely automated.” [Article: Get Started with Computer Maintenance, Microsoft Corporation, Published: September 7, 2006]*

The article goes on to describe a dozen or so maintenance actions the typical user can take. Unfortunately the list falls fall far short of the actual number of actions required to keep the typical Windows workstation running at top speed.

The Rise of the Digital Machine

Within the larger battle for business productivity and profits, the battle for workstation productivity has become one of the most important elements. This was not always the case; the problem crept up slowly. The idea of a very powerful computer that could fit on the top of a desk was unheard of a few decades ago. In the 1970's the typical business computer filled a medium-sized room and would

crush any desk. Consider the IBM System 360 Model 195 shown here, a product at the higher end of the IBM line from 1969 to 1977. IBM's product literature tells the story of its power, cost, and capacity:

> *"Main memory capacities available with the Model 195 are one-, two- and four-million bytes. The larger sizes allow users to solve complex problems more effectively and to run several programs at the same time under the control of Operating System/360 MVT (Multiprogramming with a Variable number of Tasks).*
>
> *"Model 195 monthly system rental ranges from $165,000 to $275,000, with purchase prices from $7 million to $12.5 million, depending on configuration."(IBM Archives)*

Even at millions of dollars per machine for such a tiny amount of computing power by today's standards, there were nevertheless businesses, such as banks and insurance companies, that could easily justify the high cost based on the tremendous gains in productivity. Computers today that are many thousands of times more powerful cost only a few hundred dollars each and fit easily into a student's backpack. The numbers of units are staggering; although the world's population is around 6.9 billion, we will soon be approaching two billion PCs worldwide:

> *"The number of installed PCs worldwide has surpassed 1 billion units, according to Gartner, Inc. Gartner analysts estimate the worldwide installed base of PCs is growing just under 12 percent annually. At that pace, it will surpass 2 billion units by early 2014." (Gartner Group Press Release, June 2008, referring to Gartner's report: "Forecast: PC Installed Base, Worldwide, 2004-2012")*

There is nothing to indicate the addiction to computer technology will let up anytime soon. In a 2010 press release Gartner describes the rate of shipments as follows:

> *"Worldwide PC shipments reached 82.9 million units in the second quarter of 2010, a 20.7 percent increase from the second quarter of 2009, according to preliminary results by Gartner, Inc. [Press release: Gartner Says Worldwide PC Shipments Increased 21 Percent in Second Quarter of 2010-- Stamford, Conn., July 14, 2010.]*

Windows, Windows, Everywhere

The vast majority of installed PCs are running on the Windows® operating system. Various sources tracking the operating systems of computers accessing the Internet indicate the Windows' share of the market is between 82% and 92%. A breakdown by specific version of Windows shows XP is still in the lead at 46%. (Global Web Stats, W3Counter, August 2010)

Operating Systems As of August 2010		
Rank	**Platform**	**Percent**
1	Windows XP	46.29%
2	Windows 7	18.37%
3	Windows Vista	15.95%
4	Mac OS X	7.08%
5	Linux	1.52%
6	Windows 2003	1.09%
7	iPhone OSX	0.81%
8	Windows 2000	0.30%
9	WAP	0.09%
10	Android	0.07%

Businesses of all types are clearly dependent on this technology if for no other reason than the enormous impact on employee productivity. The business applications are endless—e-mail, the Internet, word processing, database management, purchasing, inventory control, hiring, training, accounting, sales and marketing, and almost every other aspect of operations and management. Who could resist the enormous power of these small machines? The first computer to fit on top of a desk must have seemed strange, but it is just as odd today to see a desk anywhere without a monitor on top. Windows workstations clearly are, for the foreseeable future, here to stay.

The Need for Maintenance of Windows Workstations

It is widely understood that all software applications, including operating systems, have their fair share of unresolved problems. There should be no expectation of a perfect system now or in the future. Consequently there will always be a need for tools designed to make the operating system more livable and workable. One expert explains the inevitability of the situation as follows:

> *"Most applications have deficiencies. The deficiency is something you perceive the application is lacking. When enough people feel that the application has a deficiency, the vendor normally does something about it. However, many deficiencies reflect a personal requirement that the vendor will never address. Consequently, either you continue to use the application or you obtain another application that addresses the concern." [Section: "Exploring the Alternatives to Application Deficiencies," page 464 of the book: "Windows Administration at the Command Line" by John Paul Mueller, Wiley Publishing, 2007]*

Microsoft Corporation, the developer of the Windows operating system, also acknowledges the need for supplementary tools to keep a Windows system in good working order, as seen in this quote from a support page:

> *"Thank you for your interest in Microsoft Fix it. We're working hard to automate solutions to common software problems in an easy, intuitive way that is available when and where you need it. So whether you are looking for a solution in help or support content, or an error report, Fix it provides a way to apply automated fixes, workarounds, or configuration*

changes so you don't have to perform a long list of manual steps yourself." (From http://support.microsoft.com/fixit)

Microsoft acknowledges the tediousness of "having to perform a long list of manual steps yourself." The site goes on to list a variety of minor fix-its aimed at a several types of glitches, including one entry designed to "fix Windows system performance problems on slow Windows computers." The fix provided, though, is unbelievably thin. It targets just four items, leaving hundreds of others unaddressed as follows:

"Automatically diagnose and fix causes of poor system performance, such as the power saver setting, multiple anti-virus programs running, multiple startup programs running, and too many users logged on to the computer."

IT professionals who have used any number of Microsoft-provided tools for repairing Windows-related bugs will report that in most cases, after running the tool, they end up searching for additional solutions with much greater depth and power.

Understanding the Basics of the System

Business executive and IT professionals responsible for educating users about the basics of their systems and the need for proper operation and maintenance have found that the job is easier when users understand a little bit more about the inner workings of the computer. It is important to avoid the urge to turn users into computer scientists; that degree of understanding is not necessary for proper operation of the workstation, just as the driver of a car does not need to be an expert auto mechanic. On the other hand, it is important to ensure that users are not left completely in the dark.

An example of the degree of insight needed would be a student driver learning to drive a manual stick-shift vehicle. If the student receives an explanation and simple schematic of the way the clutch works and the way the gearbox works, he or she will be better able to coordinate the gas pedal, clutch pedal, and gear shifter at the same time. In the same way, when a workstation user gains a few basic insights into the core of the system, meaning just a handful of the most important components, he or she will be less likely to continue overloading the memory and the CPU with too many applications at once. They might even stop downloading and installing every piece of freeware they find on the Internet. Some examples of the types of information that will improve the user's understanding and productivity are as follows.

Definition of a computer system: A computer system consists of four major elements—hardware, software, operators, and electricity. The hardware components are the easily visible, tangible, physical objects. The software consists of coded instructions, which are actually in the form of electrical signals and patterns. Operators are people, whether users or administrators. Oddly enough, the fourth element, electricity, often explains the entire system better than anything else.

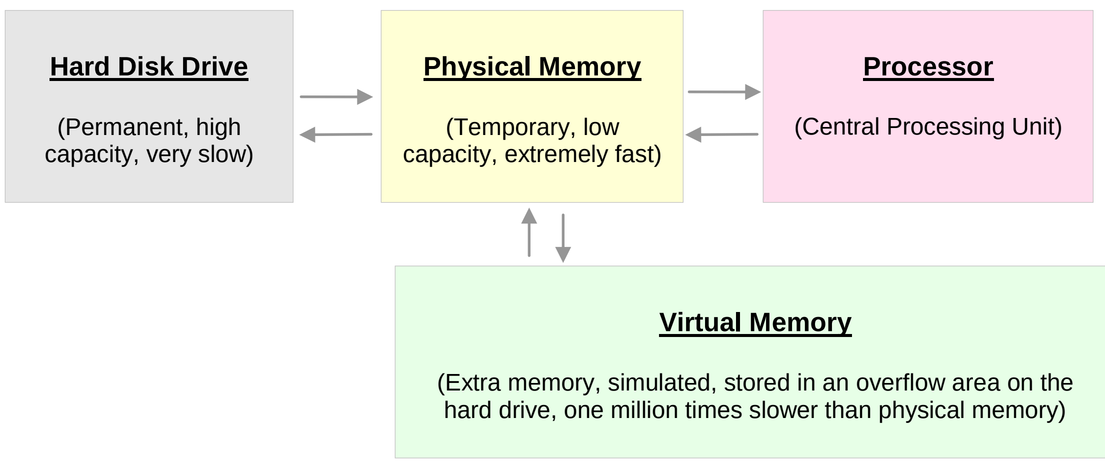
The flow of electricity: In a Windows workstation, electricity coming from a battery or a wall socket courses through the computer and lights up the display monitor. The computer's hardware consists almost entirely of static (non-moving) electronic components, essentially a collection of pathways and gateways designed to control the flow of electrical energy. The major moving components are in the hard drives, such as the rotating disk and the positioning mechanisms that place the read-write heads above the disk. Minor moving parts include the cooling fans above the CPU and near the power supply. In the electrical sense, therefore, the system could easily be thought of as "an extremely fancy light bulb." The analogy is useful only because it is very easy to see that a traditional light bulb is an electrical circuit in which energy flows through a tiny metal filament inside a sealed glass case, and that there are no mechanical moving parts, and that light and heat are generated in the process.

Pulses of energy. When bringing users up to speed on the basics of their systems, many are surprised to learn that there really are no 1's or 0's or any other types of digital numbers, or letters, or other symbols anywhere within the system. Every bit of what is called "information" in a computer exists strictly and only in the form of relatively positive and negative electrical charges. Users need to understand the hard drive stores these charges on a magnetic medium, and that the read-write heads convert these bits of stored magnetic charge into moving electrical pulses. Complex sequences of plus and minus pulses are then placed into the physical memory chip, where some are eventually pushed as "input" into the central processor. The output of electrical signals on the other side of the processor ends up back in memory and/or passes down the line into peripheral devices such as the display screens, printers, and hard drives. The cooling fans are important because all of this pulsing generates heat, just as a light bulb designed mainly to produce light will also put out heat as a side effect.

The Performance Chassis

With the above in mind it becomes easier to visualize the “performance chassis” of a typical Windows workstation computer system. Looking only at the physical or hardware aspect, and setting aside all of the software for the moment, the main chassis of the system consists of the four components having the greatest impact on computer performance and productivity: the CPU, or processor; the physical memory space (RAM), the virtual memory space stored temporarily on a special section of the hard drive, and the permanent memory stored on the balance of the hard drive. There are many other important elements in the system, but for sake of simplicity in understanding workstation performance and maintenance, just these four will be sufficient.

The basic operation or work flow occurring at the core of the system is therefore as shown in the schematic below. This illustration is intentionally oversimplified.



In this workflow, the computer’s operating system, such as Windows XP, Windows 7, Windows Vista, and so on, along with other software programs selected by the user, are loaded into the very fast but severely limited physical memory space. Physical memory exists in the form of one or more relatively expensive RAM chips. If additional memory space is required, the overflow is loaded into a virtual memory space located on the hard drive. This “fake” type of memory has no high-speed chip associated with it. Since it is stored on the hard drive, it is extremely slow compared to “real” physical memory, about one million times slower. Software programs, also known as “code” or “instructions,” are loaded into physical memory. Selected portions of the code, along with other bits of data relating to those instructions, are pushed energetically into the central processor for execution of the code.

This extremely simplified view of the core of the system allows IT professionals to emphasize to users a few of the key concepts relating to system tune-up and maintenance:

- The CPU can be overloaded by starting too many programs at once, each of which tries to push electrical energy into and through the processor
- The excessive loads will cause some programs to "hang" and the system may need to be rebooted if the CPU does not recover
- The startup and logon routine of the workstation will typically include a number of unnecessary programs, each of which places a demand on the memory chip and on the central processor
- A fragmented file structure on the hard drive can slow the system during times of heavy or repeated drive access

Grand Central Processor

The structure and function of the central processing unit (CPU) inside the typical Windows workstation is a complete mystery to many users. They usually know that there is an expensive device inside, but will have little or no idea as to how it works. Even the original designers of the chip may end up with blank looks on their faces when the processor suddenly "locks up" or freezes for no apparent reason. In this case the chip designer might grab a CD full of software and hardware diagnostic tools and start testing. A desktop user, however, might simply watch the hourglass turning over and over, as the progress bar crawls along at a tediously slow pace. It is not necessary for a computer user to take a year-long course in semiconductor circuit design, but it is important to have enough of an understanding to at least be able to determine when the system is not working the way it should. If you ask the user what the system is doing when the CPU is hanging, usually he or she will say "It's thinking," or "I don't know, it always does that when I run this program."

When users are given better insight into the structure and function of the processor, they are less likely to overload it with too many tasks at once. The first thing to have them understand is that a computer's central processing chip is nothing more than a collection of electronic pathways, all of which are conductive—allowing energy to flow—and some of which can be temporarily made to be non-conductive—to stop or redirect the flow. Energy in the form of coded patterns of electrical signals enters the chip at one end, and eventually (microseconds later) exits at the other end, with the pattern of signals having been altered in the process. One of the better ways to envision it is as a series of streets, subways, and railways in the downtown area of a large city in which cars, buses, and trains are being pushed through these pathways at incredibly high speeds. The processor, or chip, as a piece of electronic hardware, has no awareness of what is being pushed through it or why. The software, or code, in the form of patterns of electrical signals, controls the speeds and routes of travel through the city, or chip. There are no drivers and there are no pedestrians because it is too dangerous to be walking around with all that energy passing by at the speed of light. Only raw electricity can enter this city.

Users need to know that even though some processors are faster than others, with more lanes, higher speed limits, and better coordination of signals, even the latest and greatest processor can be brought to a grinding halt by loading it up with too many tasks from too many applications. Even just a few lines of bad code from one poorly designed piece of software alone can produce total gridlock by, for example, placing the processor into an endless loop of trying to solve a problem that can't be solved, while forcing it to stay on that one task alone, and commanding it to keep on trying no matter what. In the downtown area of a large city you can create the same type of gridlock just by forcing every vehicle to make a left turn at every intersection.

Physical Memory Is the Place to Be

Physical memory is where all electrical signals are temporarily stored before being pushed into and through the processor. In our downtown traffic analogy, physical memory would be equivalent to a giant parking lot full of vehicles at a baseball game. The vehicles are released from this storage area and directed into the city at regular intervals. But if they are released all at once, there will be a giant traffic jam and for several hours nothing productive will get done.

Have you noticed how pushy some software applications can be when it comes to taking over the system's resources? Every software developer wants their special application to become the star of the physical-memory stage, just as every celebrity wannabe absolutely must get into the most popular nightclubs in New York, Moscow, Paris, and Rome. The problem with physical memory, however, is that there are no professional bouncers at the entrance to keep out the riff-raff. If workstation users, managers, and IT staff are not constantly on alert, all sorts of pesky applications and sneaky add-ons will manage to crowd their way into the physical memory space. This of course interferes with the more productive applications people need to run. Users with large, essential spreadsheets for managing inventory, or for analyzing sales and marketing, must not be slowed or stopped just because some Web-surfing toolbar insists on updating itself every six minutes. Invisible intruders like these are eating away at the productivity of the system and the profits of the company. The essential job of efficient users, owners, and IT managers, therefore, is to tightly control the flow of electrical energy into the fast-acting but expensive and very limited physical-memory space. There has to be smart selectivity as to which drivers and applications are most important to user productivity. Even on a new system one of the first jobs of an administrator is to detect and immediately eject each and every unnecessary application that has managed to hitch a ride onto the new desktop.

Virtual Memory Is Slow and Fake

Because the extremely fast random-access-memory, or RAM, chip is so expensive, and because the operating system can only manage a limited amount of it, the system will sometimes run out of this type of memory. In that case, the alternative form of storage known as virtual memory comes into play. Virtual memory is simulated or fake memory. It is created on a special space on the hard drive reserved for virtual memory. The system treats this fake memory as if it were real, additional RAM. One thing users need to know is that real memory is a million times faster than virtual memory. Therefore when physical memory is full and the system has to go to virtual memory, the execution of instructions will take a lot more time. This is one more reason to keep the main physical memory chip from being overloaded.

Hard Disk Storage

A computer system runs entirely on complex patterns of electrical signals. All software is in the form of electrical signals. It is not in the form of words or sentences or “lines of instructions” as seen in a printout of what the code is supposedly doing. These lines of instructions are simply interpretations of what the code is doing. A tremendous number of wildly different patterns of electrical signals are required to run a computer. These signals must be stored somewhere. They cannot all fit into physical memory at once, and even if they could they would need to remain intact when the computer is turned off. This is the job of the hard disk drive. Because the disk is a magnetic medium, it is capable of storing magnetic charges on its surface in a way that holds them more or less permanently in place when the power is turned off, in the same way that a magnetic cassette tape stores the signals required to reproduce a piece of music. The problem with this type of storage is that the files can be arranged neatly or they can be scattered all over the place. If the storage is very neat and efficient, the data can be retrieved quickly when needed. If the files are scattered and fragmented into smaller bits, it takes much more time to collect them and move them into physical memory. Therefore, users need to know that one of the more important maintenance tasks is to run a file defragmentation routine every few weeks to rearrange the data on the hard drive into more efficient storage patterns.

The Software Controls the Hardware

A computer printer produces a printout based strictly on a series of electrical signals sent to the printer by the processor. These “output signals” are generated by the software controlling the system. Since the software itself is also a collection of electrical signals, we have a situation where one pattern of electrical signals acts as an input of energy into the processor, which then exits in a different form, as an output signal to be relayed to one device or another. Some of these output signals are routed to the display screen; other signals are routed to the printer. The entire scheme is that one pattern of electrical energy is controlling another pattern of electrical energy. The different sets of energy signals operate in a hierarchy of command, in which the signals coming from the operating system have a deciding influence over the signals that control various physical devices such as the video monitor. A basic understanding of the way these software elements work together is essential to achieving a more efficient maintenance of the workstation.

Operating Systems Are Large and Complex

The computer’s operating system, such as Windows XP or Windows 7, is a very large and complex collection of electrical energy patterns that are used to control the many activities of the computer. The operating system commands the processor, the memory chip, the disk drives, the video display, and all other devices connected to the computer. The term “configuration” of the computer actually refers to the shapes of these patterns of electrical energy. It can be very helpful to users to differentiate between the physical or hardware elements and the software, or electrical-energy-pattern elements. A hard disk drive physically has a definite shape and size which does not change. When the disk is used for storage of programs and data (both of which are forms of electrical energy), the shapes and patterns of electrical charge will vary, but the physical structure of the disk will not change. The same is true of random access memory chips and the processor. The hardware aspects are physically stable, while the software (electrical pattern) elements are changing the entire time the computer is up and running.

The important thing for users to understand about this view of the system is that the maintenance of the computer really refers to the maintenance of the patterns of electrical energy running through the system. This explains why an electronic device, which has no moving parts, can rapidly degrade in terms of performance despite the fact that no degradation of the physical structures has occurred. It also explains why a brand new computer, fresh out of the box, may immediately need tuning to correct the inefficient patterns of electrical signals already set up within the system.

The Problem of Device Drivers

Drivers are an important part of the operating system. Device drivers are relatively small bits of software code designed to control the connected devices such as the mouse, keyboard, printer, monitor, disk drive, sound card, and so on. Driver errors can have a serious impact on user productivity. When a driver is not properly configured, it cannot correctly interpret the signals coming from the rest of the operating system. This can result in system slowdowns, freezes, and crashes. It is not unusual in many companies to have users losing several hours of production at a time because of device driver problems.

The Big Bad Windows Registry—To Fear, or Not to Fear?

The Windows registry! For some reason it strikes fear into the hearts of users and technicians alike. It's hidden and it's mysterious. It's as if no one knows what goes on in there, but they do know it's very dangerous, and that only "advanced users" should ever go near it, or you could end up with a *blue screen of death!* That can't be good. The registry has the same reputation as a locked room at the end of an unlit hall on the third floor of a creaky, old haunted house. You've never been in there, but you've heard stories. You *could* go in there if you wanted to, right, because they don't scare you, do they? Except you would need a good reason to go in there, and you just don't happen to have one at the moment, and there are much better things to do with your time, right? But you definitely could go in there if you wanted to.

Now how would you feel if you finally did go in there one foggy evening, with a heavy flashlight in one hand and a sharp kitchen knife in the other, and there were no fiery demons or mummified corpses, only rows and rows of dusty old bookshelves and a sleepy old librarian filing away at large piles of tiny data cards, placing each carefully into one of fifty small wooden drawers? The registry that we all thought was a torture chamber turns out to be nothing but a library, a database containing the tens of thousands of settings needed to run the computer. There are records defining such things as which applications to use when opening certain types of files, for example, using Word to open .doc files, Excel to open .xls, Outlook to open .pst files, and so on. The registry turns out to be very well organized. It has to be, because the operating system is accessing the registry thousands of times per second to make decisions about what to do next. For example, when a user logs on, the registry delivers detailed data about the user's preferred settings, which helps the operating system set up the desktop. Another section contains the security settings for the computer, helping the system control access to files, applications, and networks.

It sounds like a good way to make the operating system more efficient, by keeping all of the configuration settings in one large database where every piece of data has an exact format and location. There's only one problem: it is nearly impossible for any database of any size, on any subject, to be accurate and up-to-date at all times. Every database collects errors. When an error occurs in the Windows registry, it might be innocuous and go unnoticed, or it might cause slowdowns as the operating system struggles to find the correct way to execute a command. As the number of registry errors increases over time, the chances for random slowdowns also increases. Since the registry is so large and so critical to the operation of the computer, it is essential that it be analyzed and cleaned up on a regular basis. The development of intelligent tools for correcting the registry is an important industry all by itself.

User Applications

At the machine level, all hardware devices are controlled by drivers. These device drivers receive their instructions from the operating system. If all you did was turn on the machine and load the operating system, the computer would run but it would not accomplish much. It would be like starting a car but not driving anywhere, a waste of gasoline. To accomplish work, you have to load an application, a piece of software dedicated to some kind of task, for example, loading Outlook to receive and send e-mails, or Word to write a report, and so on. The problem with applications is there are so many of them, and they all behave differently. Ideally the application has been written in such a way that it is fully compatible with the operating system, and so it won't crash the system. But that does not mean the application is fully efficient. It can be very well written but still leave fragments of code lying around in memory, or it can create files that will slowly clog the hard drive, and so on. It's a lot like the wear and tear that occurs when you start up a car and drive somewhere. Over time, the car collects more and more debris and eventually needs to be cleaned out. The problem is worse for some users, for example the power user who downloads and installs new applications on a regular basis, each of which adds to the registry, places news loads on memory, and increases the number of files on the hard drive. Computers are meant to be used, just as cars are meant to be driven, but the more they are driven, the more important it is to perform maintenance and avoid costly breakdowns.

Is It a Malware Infection or Is It a Configuration Error?

How often has a workstation user complained that his or her system was infected when in fact it turned out to instead contain one or more configuration errors? Consider the following statement from Microsoft illustrating the way configuration errors can present the same symptoms as computer viruses:

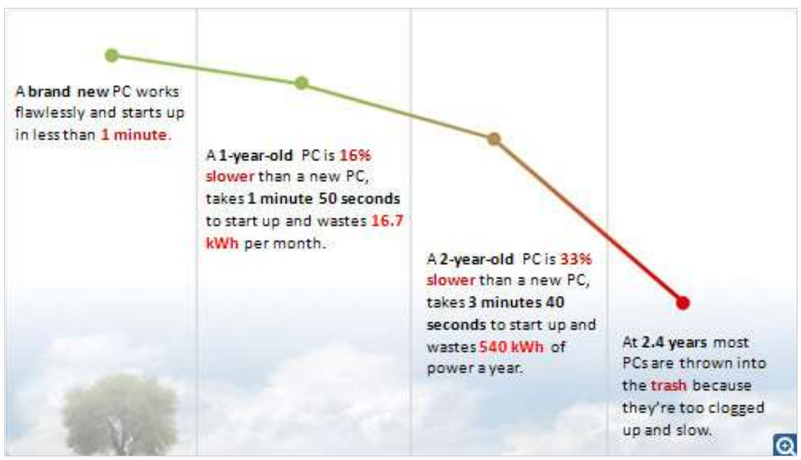
> *"The following are some primary indicators that a computer may be infected:*
>
> - *The computer runs slower than usual.*
> - *The computer stops responding, or it locks up frequently…*
> - *You see unusual error messages.*
> - *You see distorted menus and dialog boxes…*
> - *New icons appear on the desktop that you did not put there, or the icons are not associated with any recently installed programs…*
> - *A program disappears from the computer even though you did not intentionally remove the program.*
>
> *"Note: These are common signs of infection. However, these signs may also be caused by hardware or software problems that have nothing to do with a computer virus." [From KB129972: "Computer Viruses: Description, Prevention, and Recovery," by Microsoft Corporation]*

Bad code is bad code, whether from poorly written software or intentionally malicious viruses, or just the result of incorrect settings by users. Mistakes in the operating system, the Windows registry, and in other applications must be detected and corrected at every possible opportunity.

Performance Degradation Over Time

In interacting with users in all types of companies, iolo has observed that the vast majority are well aware of the fact that computer performance tends to degrade over time. Some users, however, are somewhat mystified by this fact because the workstation appears to have "no moving parts" and everyone knows an electronic device does not wear out the way a mechanical device would. What users need to understand is that while the hardware is relatively stable and static, the *configuration* of the machine is dynamic. That is, the electronic settings and configurations are changing rapidly while the machine is in operation. It is the dynamic settings that directly impact performance, for example by sending too many signals at once into memory and the processor, causing traffic jams, or by storing files electronically on the hard drive in a disorganized way, or by adding more and more applications to the startup routine, eating up space and time and energy. The maintenance of the hardware, therefore, is not so much an issue as maintenance of the configuration.

Studies conducted by iolo Labs have confirmed and quantified what people have already seen on the job. The graph below illustrates the amount of performance degradation that occurs over a few years' time in the typical workstation.



As can be seen, the startup time is becoming slower and slower. The startup time is one of the more important features for PC users. (iolo Labs Research Study—Windows Startup: Observed Changes Over Time, iolo Labs, published June 19, 2009)

CPU resources are also affected by this kind of performance degradation. The same study found that by the time the PC had been in use for 24 months, the majority of its increased CPU load was attributed to the impact of unneeded startup tasks.

User Types and Styles—the Gambler and the Scientist

There is a broad spectrum of differing attitudes about what are the proper levels of maintenance and tune-up for equipment such as bicycles, lawnmowers, water heaters, and Windows workstations. At one end of the spectrum is “The Gambler,” and at the other end “The Scientist,” with the majority of managers, and IT professionals falling somewhere in between. The Gambler takes advantage of the fact that you never really know whether or not a system will break down and when. It might go on for years with no trouble whatsoever. Therefore, to his or her mind, it seems reasonable to “play the odds” and hope that many of the recommended maintenance actions aren’t really needed. This is further justified by saying, “If something were to go wrong, we could always fix it after it breaks.”

The Scientist on the other hand says, “Out of a given number of cars or workstations, we know for a fact that a certain percentage will experience various levels of major or minor problems at various intervals.” The Scientist, being a more level-headed style of manager, and averse to life’s little dramas, does not want to go through the more expensive cycle of breakdown and repair if he or she doesn’t have to. The preference is for the most uneventful workday possible. The Scientist actually takes great pride in controlling events in such a way that nothing bad ever happens. The opposite-minded Gambler takes pride in not worrying so much, and the ability to let things ride, and to “roll with the punches.” The Gambler thinks the Scientist is perhaps too fearful, controlling, and overly cautious, while the Scientist thinks the Gambler is far too reckless with company resources. In the end it’s arguably just a difference in style and personality, and it’s up to each individual to decide where he or she falls along the preventative-maintenance spectrum. You could ask yourself, “Which is better, the availability of a world-renowned cardiac surgeon guaranteed to be at your side after a severe heart attack, or a nutritionist and trainer who starts you out at an early age on the best preventative diet-and-exercise practices?” The Gambler looks at the odds and says “I’ll take my changes and enjoy my life of steak and eggs.” The Scientist says, “I’d rather not gamble with such critical resources as my body and my life, and so I will take the safer course and reduce the chances of a life-threatening event by practicing exactly what the Science of Probability so clearly teaches us.” Either may be right or wrong, but each following their chosen path will die happy regardless.

At iolo we have determined that we can’t do much about these differing styles of management, and we cannot prove with total certainty which is right and which is wrong, except to say to the Gambler that the setup time and costs associated with proactive maintenance and tune-up are far lower than one might expect, and so there is no downside to the approach of the Scientist. The tools needed to prevent unnecessary downtime and increase workstation performance are available to both ends of the spectrum. The methods to set up and accomplish it with very little effort are discussed in greater detail in the later sections of this report.

The “No Moving Parts” Argument

Every time you turn on your PC, the first thing it does is conduct a Power-on Self Test (POST) to determine whether or not the processor, memory and other critical components are working properly. If everything is OK, you will hear one short beep at the end of the test. Check it out the next time you boot up. If there is a problem, however, you will hear more than one beep. For example, in some systems a processor failure is indicated by five beeps. But unless your job or your favorite hobby involves the attempted revival of dead computers, how often in your entire lifetime have you heard more than one beep? OK, to make it more relevant, how about in the last ten years? Yes, in the early days there were far more hardware failures, but that was before quality control levels in the electronics industry reached astronomical heights and stabilized there. Most people will say, out of thousands of computer startups, “Never more than one beep, or almost never.”

It is for reasons like this that many people have come to believe that electronic devices such as PCs and laptops do not need regular maintenance. They typically say, “There are no moving parts, so what can go wrong?” It’s true, to a degree, that as long as the component is not subjected to heavy abuses such as higher-than-specified voltages or excessive heat, such as in a CPU fan failure, the component is not likely to fail within the normal lifetime of the computer. But they are forgetting one thing—the *physical condition* of the hardware is not the most important factor in Windows workstation performance degradation. It’s the *software configuration* of the device that causes the major problems. The hardware itself may be stable and capable of passing all circuitry tests. But the configuration of the device, the internal software settings that guide the flow of electricity through the system, are invisible. It takes special tools to detect these settings, to uncover inefficient configurations.

Let’s say, as commonly happens, the processor is being overloaded with too many demands at once. We see this in the form of a large Excel spreadsheet freezing the system until a recalculation cycle is completed, or in a database query that takes ten minutes (or several hours) to finish processing. During these times, all other processor tasks have to wait, and this in understandable. But what about a processor that runs slowly all the time without such demanding applications in play? What if the user has been adding new software applications every so often and some of them are demanding more and more memory and processor cycles each time the computer is started? Now we have created an overload condition with no easily visible cause. The user is running only e-mail, Internet, and word processing, but everything loads slowly and runs slowly. Why? Because the proper configuration of the system has not been maintained. The hardware may be physically intact but the configuration has degenerated in a way that impacts the performance of the entire workstation. The concept of "maintenance" is most closely associated with keeping physical objects in good working condition. It is not as easily associated with keeping a device properly configured.

Maintenance Induced Damage

People responsible for the maintenance of critical equipment in a wide range of industries are well aware of the fact that more damage can be caused by maintaining the equipment than by leaving it alone. It's considered dangerous and they call it "maintenance-induced damage." A typical example is in a hydraulic system where the pump is fully broken-in and running smoothly, with all mating parts properly sealed and lubricated. All you have to do to ruin the pump in this case is open it up, replace a few seals, re-grease the mating surfaces, and put it all back together. Suddenly the pump is leaking, running hot, blowing fuses, and shutting itself down. What happened? It's simple. The act of routine maintenance has disturbed the equilibrium of the system. Auto technicians are full of such stories. Ask any one of them if they have had a customer recently who tried to fix his or her own car. If so, ask what kind of damage they did and what it cost to repair. The problem occurs in commercial aircraft applications as well, where maintenance manuals specifically warn against even just opening and inspecting a device before its scheduled overhaul period. The concept of maintenance-induced damage is the basis for that old and wise saying, "If it ain't broke, don't fix it."

Many IT professionals apply this same principle to their servers and network devices and configurations. It takes hard work to find exactly the right settings for these systems. Once that occurs, unless you have a very good reason to do so, you should never, ever, disturb a working setup. But does this same principle apply to user workstations? The answer is no! The reason it does not apply is the typical user workstation is not under very tight control the way servers and other network devices are. Users and their many applications are doing disturbing things to desktop systems every hour of every day. It's only a matter of time before one of these actions, or the accumulated effects of many such actions, causes a noticeable slowdown or a total breakdown of the system. Therefore, you could amend that old saying to state the case more accurately: “If it ain’t broke, and if no one is messing around with it, you can leave it pretty much alone; but if it is in the hands of one or more uncontrolled users, you better check it out now and then and maintain it before it crashes.”

The System Mechanic® Philosophy

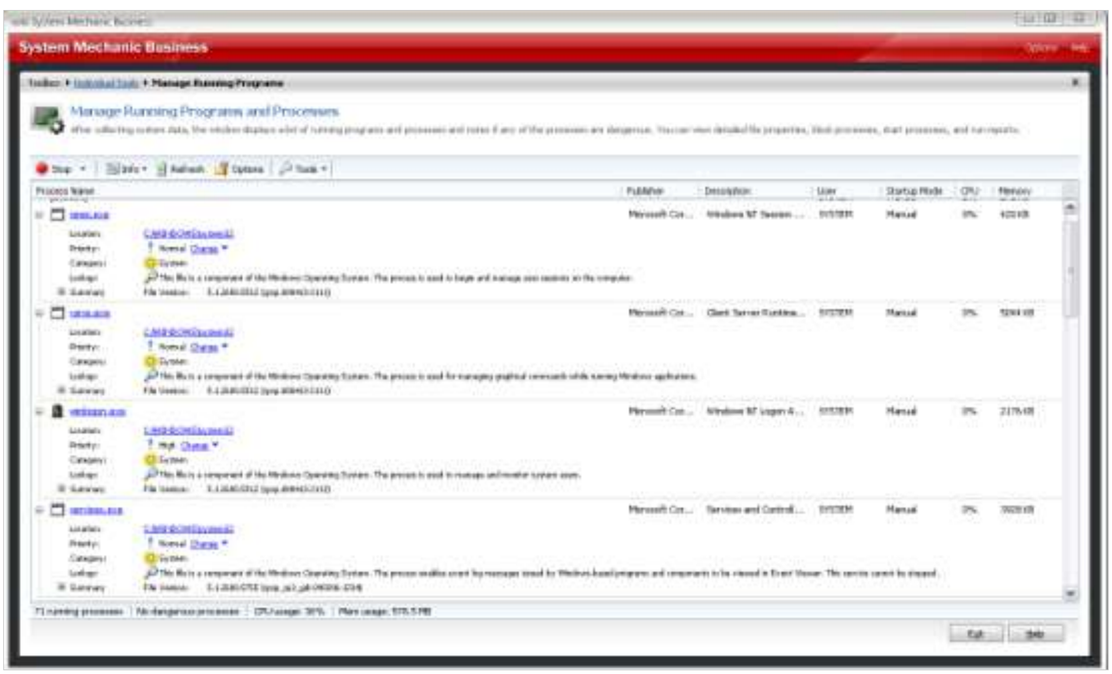
Based on our experience in home and business environments over the past decade, we can safely say that the most effective solution for workstation maintenance, tune-up, and repair is iolo’s System Mechanic®. A look behind the scenes at the iolo mission and product-development philosophy will clarify this position. The primary driving force behind the development of System Mechanic is iolo’s firmly held belief that computers are meant to speed things up, to make working sessions more productive, and life in general more entertaining, to facilitate personal and business communications, and to affirm a sense of interconnectedness. We view the Windows workstation as an ally in all of these endeavors, and therefore we have focused our efforts completely on this operating system.

Anything that gets in the way of the ability to use Windows systems efficiently is an enemy to be defeated, a technical or ergonomic challenge to be confronted and resolved by iolo. Because of this highly specialized focus, we are able to go to great lengths and to delve deeply into the Windows arena, searching for all sorts of complications and coming up with new ways to solve each of them.

As a result, probably the best way to understand the System Mechanic philosophy is through the comprehensiveness of the solution. Here the word “system” refers to the complete set of key components comprising a Windows workstation. In brief, on the hardware side, System Mechanic addresses the loads placed on one or more central processors, memory chips, and hard drives. On the software side it detects and repairs the critical settings relating to the operating system, Windows registry, startup programs, running processes, and so on. It locates and resolves security vulnerabilities, inefficient Internet connectivity settings, and errors in the way that data and programs are being stored on disk. The goal of iolo technologies is to provide the most effective methods for accessing, monitoring, and controlling every aspect of the system having any significant impact on the productivity of the workstation or its user. Because the Windows system is extensive and complex, there are more than fifty different tools contained within the System Mechanic suite. A listing and explanation of each would be far beyond the scope of this paper and these tools are discussed in greater detail anyway in the literature found on the iolo Web site. Instead a few examples will serve to illustrate the approach iolo has taken towards maintaining, optimizing, and repairing the millions of Windows workstations used throughout every small, medium, and large enterprise.

A Sudden and Mysterious Slowdown

At the heart of the iolo philosophy is the concept of empowerment. Those responsible for maintaining Windows workstations must be armed with tools capable of penetrating into the core of the operating system, that is, beyond the level normally permitted by Windows utilities alone. Otherwise the business owner or IT manager is at the mercy of the operating system vendor, like operating a fleet of delivery trucks with all of their hoods welded shut. For example, consider the following all-too-common example of a user found “hourglassing” at his or her desk, staring blankly at a screen where nothing productive is happening. The call for help comes not from the user but the company’s owner, who somehow seems to know the minute the company’s dollars begin flying out the window. She takes action immediately: “Tom, go and check out Sally’s desktop, she hasn’t answered a single e-mail in the past forty-five minutes.” You rush over to check it out and the first thing you see is the screen is half frozen and every attempt to switch applications goes



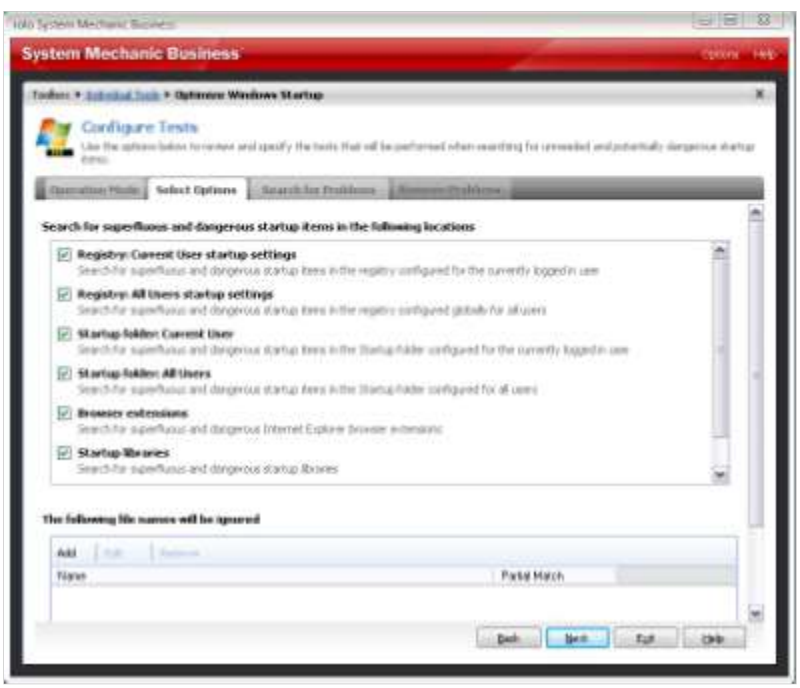
nowhere. Earlier that morning the system was fine and running at a normal speed, according to Sally, so what could have happened? A few clicks into System Mechanic sheds some badly needed light on the situation. The Process Manager™ tool shows 71 running processes, along with a brief description of each. Scanning the list you spot a few that are unfamiliar. With the built-in research tool you quickly determine which are vital and which are not needed. With just a few more clicks—running in expert mode, of course—you save a snapshot of the running processes, kill the offenders, and simultaneously prevent them from starting up in the future. A glance at the screen shows the system is now responding normally and Sally, no longer in hourglass mode, is eager to get back to work. You make a note to launch a few additional tools later to automatically monitor and streamline her system.

The process so far has been fast, reliable, and totally transparent. In iolo's view this is the way it is supposed to work when something goes wrong. The tool has to be fast, strong, and professional-grade. The ratio of productivity-to-maintenance on a workstation needs to be kept at 99:1 or better. This level of efficiency is crucial to any competitive business, but we all know how quickly things can get out-of-hand. Imagine the user is a hobbyist who enjoys tinkering with copiers, printers, and other malfunctioning electronic devices. Every company has at least one of these, who can be identified by the fact that he or she will not hesitate to spend two or three hours of trial-and-error on company time attempting to get one device back up and running. It's a nice break from an otherwise uneventful day, right? Owners, managers and IT staff are therefore well advised to keep a sharp eye out for people who love to wrestle with strange and mysterious glitches. While it is commendable that they are taking an interest and trying to help out, in the end it's the owner who is paying for the inefficiency and the lost production time, which is never good for profits.

Preventing Problems Before They Start

Business executives and IT professionals are keenly aware of the differences between proactive (disaster prevention) and reactive (disaster recovery) procedures. Business schools have entire wings devoted to the benefits of a proactive management style, as opposed to the much smaller buildings devoted to reactivity. The premise of the proactive approach—analyzing a system in advance to discover its weaknesses and then controlling the system accordingly to prevent failures—is difficult to contest. You would have to say, "Don't bother with all of that analysis, let's just deal with whatever happens after it happens," which seems a little chaotic. But let's be realistic: in most businesses, the important advantage is not simply the chance for a more orderly existence; the real virtue, the only one that matters, is cost. That is, preventative policies and actions, to be useful at all, must be significantly more cost-effective than the alternative of just letting things ride. This is the philosophy behind many of the solutions in iolo's System Mechanic suite. One of the best examples of the cost-effective prevention of problems is the Startup Optimizer tool. Everyone is familiar with the costs associated with handling a system slowdown after the fact, as in the case of Sally's workstation discussed earlier. From the moment the system first stalls or freezes, that employee is still on the clock but is not being productive. The interruption to management to detect and follow up on the situation also comes with a cost, and so too does the deployment of an IT resource. No one is working for free. Even when freezes like these happen only occasionally, they can quickly add up to hundreds or even thousands of dollars per workstation per year. Therefore the idea behind iolo's Startup Optimizer is two-fold: first, by optimizing the startup configuration, the workstation will boot faster, so that employees can get to work quickly. Secondly, by streamlining the loads placed on the system memory and processors, it allows applications to run faster from the start and throughout the entire workday, with fewer chances for conflicts and breakdowns. The net effect is enhanced productivity and accumulated savings. As stated earlier, though, for this type of proactive approach to be truly viable, the process of optimizing the system before trouble strikes has to be fast, accurate, and cost-efficient.

The way it works is the manager or IT administrator optimizing the workstation first chooses from the Quick, Deep, or Custom optimization modes. The Custom mode provides access to a number of individualized settings, allowing complete visibility



into, and full control of, the areas of the system to be analyzed and repaired, as shown in the screenshot above. The tool operator is further informed by the explanations that accompany the items on every screen, iolo's way of distributing in context the intelligence accumulated over the years, concerning key aspects of the Windows system. By clicking "Next" to start the "Search for Problems" phase, the tool takes advantage of these insights by digging into every sector from which startup programs are being launched. Many users are surprised at seeing the number of launch locations embedded in the Windows system.

The tool then evaluates everything that has been found and reports on the status of each item along with iolo's recommended actions. The wealth of accumulated intelligence is what makes it possible for anyone to completely optimize the startup configuration within a matter of minutes. As with each of the other tools in the System Mechanic suite, the operator has the chance to review and customize the repair actions to be taken. With the startup configuration under much better control, the workstation will boot up and run far more efficiently and the chances of a costly slowdown or freeze are greatly reduced.

Visibility, Trust, and Control

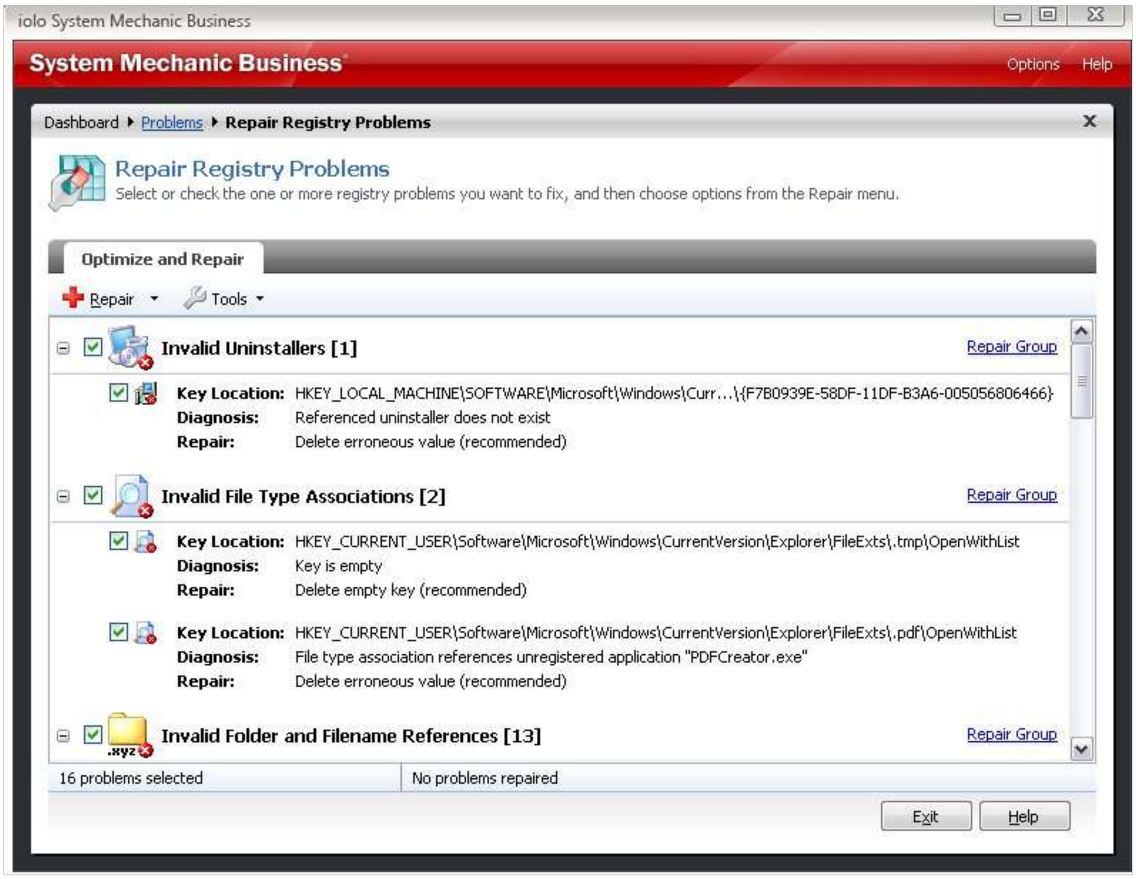
There's a reason System Mechanic solutions have been developed in this way, with strong visibility into the workings of the tool. None of iolo's tools have ever been developed in a vacuum. On the technical side alone it would be impossible to do so, since these tools have to work intimately within an intricate web of operating system complexities. The issue of compatibility alone forces developers to think very clearly about a great number of technical details and the operating system and user interactions. Every line of code must be tested extensively until it proves itself capable of running perfectly on each of the Windows platforms.

On the human or ergonomic side—dealing with the way different people interact with their Windows workstations—the tool needs to accommodate a broad range of considerations, and especially the core issues of visibility, trust, and control. Owners and managers responsible for maintaining company systems are naturally wary of any tool that probes deeply into the engine, transmission, and every other critical area of the workstation. IT executives would not be doing their jobs as responsibly as they should if they were not very cautious before allowing a tool to operate on these systems. At iolo we recognize this as just another type of compatibility issue. Call it the "management-style" requirement; it is every bit as important as any major technical issue. Therefore iolo has worked very closely with these concerns during the development and refinement of the System Mechanic solution. One of the most consistent observations we have made in these interactions is that, while some executives require very complete insight into the solution, others prefer not to see every detail, or none at all. They somehow sense who and what to trust, usually relying on the opinions of experts, or on word-of-mouth from peers and people they know and trust. One reviewer who falls clearly into the "just-fix-it-for-me" camp expresses it this way:

> *"I like the System Mechanic interface: it's easy to use and effective. You don't have to know much about the interior logic of your PC; just do a deep scan and let the software automatically fix problems it finds. (If turning over control makes you nervous, you can opt to select the fixes you want to implement. Me? I just clicked "fix all" and crossed my fingers.)"-- New York Times, Gadgetwise Article: "An Easy, Painless Way to Keep a PC in Shape," by Rik Fairlie, September 14, 2010*

Many IT professionals are understandably more skeptical and need to be more engaged in the details. They do not normally trust a solution unless they can see exactly what it is doing and why. The position iolo takes on the matter is that both styles must be accommodated. Consequently the tools in System Mechanic are designed with the ability to toggle rapidly between the high- and low-visibility modes. This is especially important when working in any of the numerous areas considered highly sensitive or crucial to workstation stability, including the placement of heavy loads on processors, managing physical memory, repairing the Windows registry, fixing security vulnerabilities, adjusting network connectivity settings, and the configuration of file-storage on the hard drive.

A common example is seen in working with the Windows registry. IT professionals routinely want to know which types of entries in the registry are being detected as errors and why. The way System Mechanic addresses this issue is by allowing the administrator to view each of the items found during the tool's analysis of the



registry, with complete details as shown in the screenshot above. The operator then has the option to accept or reject individual actions before they are implemented. By engaging closely with the tool in this way and seeing exactly how it works, those responsible for workstation performance can build the required levels of familiarity and trust and eventually feel comfortable with letting the tool run at top speed, or letting it run automatically and unattended on a regular basis. The trust-development process is crucial to attaining the most cost-effective levels of proactive maintenance.

The Impracticality of Manual Maintenance Methods

Every once in a while in a small, medium, or large business, we run across an IT manager attempting to maintain a fleet of workstations using a combination of manual and semi-automatic methods. It is difficult for most people to appreciate the monumental nature of this task and the level of dedication it requires. Even small offices managing just three workstations find it nearly impossible to perform these tasks on a regular basis. Consider the following attempt using an Excel spreadsheet to track the company's systems and key maintenance actions. Only a partial list of the maintenance actions and a sampling of the workstations are shown.

	Workstations										
Maintenance Actions	Exec 1	Exec 2	Fin-ance	Acct 1	Cust. Svc 1	Cust. Svc 2	Sales 1	Sales 2	Prod. 1	Prod 2	Etc.
Repair and backup the registry											
Delete junk files											
Streamline the startup process											
Repair/defrag all hard drives											
Optimize Internet settings											
Fix security vulnerabilities											
Create emergency boot CD											
Securely delete sensitive files											
Customize the Windows settings											
Repair low memory problems											
And so on ….											

It seems simple enough. All the administrator has to do is run through the checklist on each workstation, wait a few days or a week, and repeat the same routines. There are a handful of problems with this approach, however, as illustrated by a few key questions: Does the IT manager or his delegate have the time to perform these actions manually? How long does it take per workstation, and how much does it cost per maintenance cycle? How many different software tools are being used, and from how many different sources? Are any of these tools automated? If so, what impact does the automated deployment have on the workstation's availability and on the user's productivity?

What iolo has found over the years is that no one in any organization ever has the time to run a set of procedures like this repeatedly and consistently. IT resources are inevitably called to other duties and the system soon falls into neglect. Even if this were not the case, the economics of the approach are not sufficiently viable. Skilled human resources and personal time and attention are expensive commodities. While workstation maintenance is critical to business success, there are limits to the resources that can reasonably be expended. We would like to say it is worth every minute and every dollar, no matter what, but we would not last long with that argument, as it is not very realistic. The only way to complete these tasks as often as they need to be done is with an automated solution. It can't be just a "simple automation" where a tool is set to run "every so often." It has to be far more sophisticated and intelligent than that, or it will end up creating a new set of problems having to do with the commandeering of system resources and the interruption of the user's productivity. The idea of a systematic approach to workstation maintenance, cycling through the list of actions repeatedly and in a way that does not drain the pool of human resources, and which does not create new problems in the process, is one of the core considerations in the development of the ActiveCare solution, as will be discussed in greater detail in the following sections.

Protect Your Best Producers—Never Make Them Wait

The number one enemy of productivity is *waiting.* Waiting for supplies or instructions, waiting in lines, waiting for crooked politicians to be thrown out of office, or standing idly by while the computer chugs away at some unknown background task—no other force can put such a mind-numbing chill on a company's ability to produce at high levels. Everyone hates waiting. Therefore one of the most important tenets of the iolo philosophy is, "*Never interrupt a productive user, and never make them wait."* Unless there is an unforeseen breakdown, usually because preventative measures are non-existent or have failed, there is never any reason to stop a workstation user during the heat of production just to perform a maintenance task. On the contrary, iolo has consistently found, by the intelligent use of digital resources all needed maintenance and tune-up tasks can be completed easily within the naturally occurring periods of processor inactivity that every Windows workstation experiences throughout the working day.

There are two types of waiting, *necessary,* or unavoidable and related to productive activities, and *unnecessary,* or wasteful, preventable and not related to any productive activity. A user having to wait for a large spreadsheet to calculate is engaged in a productive and necessary action. Some relief could be gained by running a tool to reduce the load on the processor before working with the spreadsheet, and sometimes this could be a very significant amount of relief. But beyond that, the time it takes to complete the work is largely unavoidable. This type of waiting does not cause the most frustration. It's the other type, the waiting a user has to do while nothing in particular is running on the system, or at least not anything the user had chosen to run. This type of waiting is the result of a configuration error. One of the most common examples is waiting for an update to be completed on the system. This happens when Windows update is running in automatic mode and it is allowed to occur at any time of day. In this case Windows is constantly checking for updates and when it finds some, it downloads them and begins installing them, pretty much regardless of whatever else the workstation might be doing. If the updates are substantial, the system can be kept busy for a very long time, and then it will often need to be restarted afterwards.

Although it is critical to keep a Windows workstation up-to-date in this way, many users in response to these slowdowns end up simply turning off automatic updates, once they discover it is the source of the slowdown. At the same time they do not then follow up regularly to run the updates manually. This of course is not a good solution and it can lead to an insecure Windows system. A much better solution would be to design the update service in such a way that it is sensitive to the usage of the workstation, and program it so as not to interrupt the user's work. This is easy to say, but not so easy to do. It would take some nicely designed subroutines to detect all of the relevant workstation conditions and to take them into account during the update operation. This idea of never interrupting a workstation user engaged in a productive activity is another of the major drivers behind the development of iolo's ActiveCare solution.

The Slow and Steady Corrosion of Workstation Performance

Workstations are under attack on several fronts, each threatening the productivity of the business. On the one hand there are malware producers who are launching thousands of bits of destructive code aimed at every vulnerability in the Windows operating system. On another front there are the many different types of system configuration errors, a great many of which will have an impact similar to that of a malware infection, slowing down the system and reducing user productivity. These configuration conditions are not static, and so there is no simple, one-time solution. Fighting the battle against these elements therefore requires constant alertness as well as better and better wartime intelligence. On the anti-malware side, iolo and others have taken the approach of identifying threats on an ongoing basis by the unique signatures found within the malware code, and using these signatures to detect and destroy them on Windows systems. In a similar way, on the configuration-error side, iolo technologies has developed its unique Tune-up Definitions™ database, a highly intelligent and methodical approach to detecting and correcting more than 12,000 system-configuration errors. But as in the case of malware, this is not a one-time fix. As new types of errors are discovered, they are added to the Tune-up Definitions database and these updates are distributed to System Mechanic customers for immediate implementation. Therefore it is an organic, living and growing approach to keeping Windows workstations in top-notch condition. For this system to work, however, users need to perform automated scans on a regular basis to check for old and new types of errors.

The approach can be compared to the way the Golden Gate Bridge in the San Francisco Bay is constantly inspected and maintained. Many people have the misconception that the paint, intended primarily for corrosion protection, is entirely replaced every year, or every so many years. While it is true that the entire bridge is inspected carefully over and over again, the repair of steel parts and the touchups of paint are performed only in those areas where corrosion is found.
In a similar way, ActiveCare is designed to perform only those repairs and optimizations specifically needed by the PC at the time it runs. It does not waste time "going through the motions" like ordinary schedulers that will spend hours performing operations simply because they were scheduled to be done. These two concepts, of constant alertness for signs of performance corrosion, and making only those repairs that are truly needed, are important elements in the design of the

Seventeen ironworkers and thirty-eight painters battle wind, sea air, and fog, often suspended high above the Gate, to repair corroding steel. Ironworkers replace corroding steel and rivets with high-strength steel bolts, make small fabrications for use on the bridge, and assist painters with their rigging. Ironworkers also remove plates and bars to provide access for painters to the interiors of the columns and chords that make up the bridge. Painters prepare all bridge surfaces and repaint all corroded areas. (Golden Gate Bridge Highway and Transportation District)

ActiveCare solution. It is this level of intelligence and efficiency that makes the solution viable in business environments where entire fleets of computers must be kept up-to-date and running smoothly without the inordinate consumption of expensive human resources.

The Magic of ActiveCare®

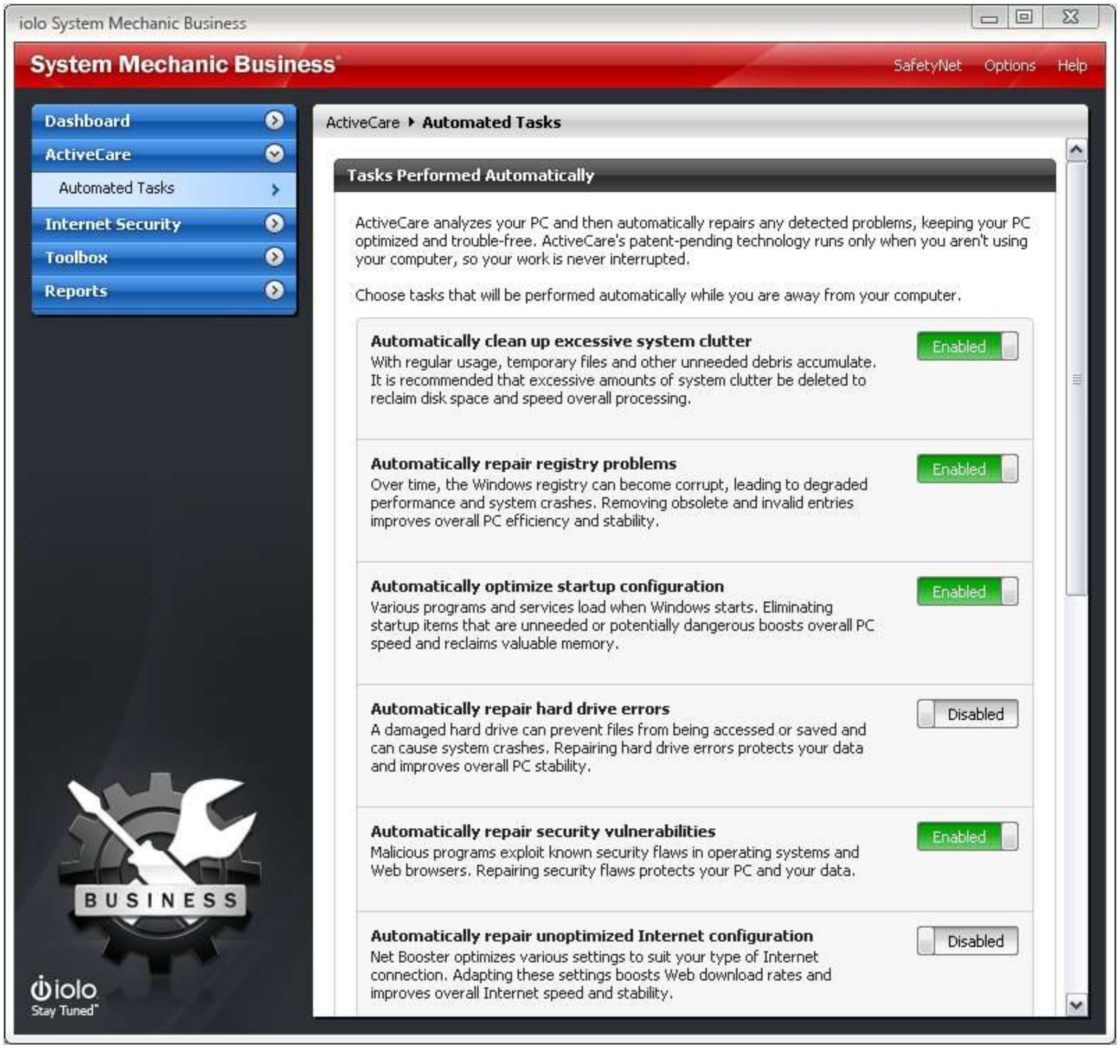
A few of the key qualities of ActiveCare have been discussed in the above sections, but what is it exactly and what does it do? ActiveCare is iolo's highly innovative, patent-pending solution for performing a complete list of the most important workstation-maintenance-and-repair operations in a fully automated fashion, and in such a way that the user of the station is never interrupted, and so that zero system resources are consumed. If you think about it, this sounds like some sort of magic, and to many skeptical minds in the business enterprise it actually is just a little too incredible. While we would like to have, or at least claim to have, powerful magical abilities, we have to admit that it's really nothing more than very hard work, resulting in our most intelligently advanced workstation-optimization technology to date.

One famous author explains the illusion this way:

> *"Any sufficiently advanced technology is indistinguishable from magic."*
> *(Clarke's Third Law, from "Profiles of the Future,"1961, by Arthur C. Clarke, English physicist and science fiction writer.)*

As it turns out, in corporate business environments, owners and senior executives are constantly looking for magic anyway, in the form of outstanding results. They always want the highest levels of production with the least amount of effort, plus low costs and top quality, all at the same time. If that's not a request for magic, nothing is. But they also know that you have to work hard for it, and you have to use all of your smarts, which is exactly what iolo has done with ActiveCare.

The first thing ActiveCare does to get big results with little effort is to take over the list of maintenance actions that should be run on every workstation as discussed earlier, ideally by setting them up to run automatically on a regular basis. No more maintenance checklists or spreadsheets, no more forgetting to run the list of tasks, no more reminders in the calendar adding to the IT manager's weekly headaches. He or she can just enjoy the benefits while ActiveCare does all of the work. A sampling of the types of actions that can be automated are shown in the screenshot below. Each automated task is enabled or disabled with a single click.



Once the list is enabled, ActiveCare technology takes over, checking for erroneous and non-optimum settings in all critical areas of the system.

Instantly Appearing and Disappearing at Will

One of the most important qualities of ActiveCare, the one that makes it seem most ghostly and magical, is iolo's patent-pending idle-time processing. This technology is able to detect when the system is being used by the operator or a running program, and thereby never interrupting the system in use. This is not simply a timer that counts from the last time you hit or key of clicked the mouse. The program, more sophisticated than that, detects when an application is running in full-screen mode, and senses as well the load on the processor. For example, using the default settings, ActiveCare maintenance actions will not begin until the system has been idle for at least twenty minutes, and only if the CPU usage is 25% or less.

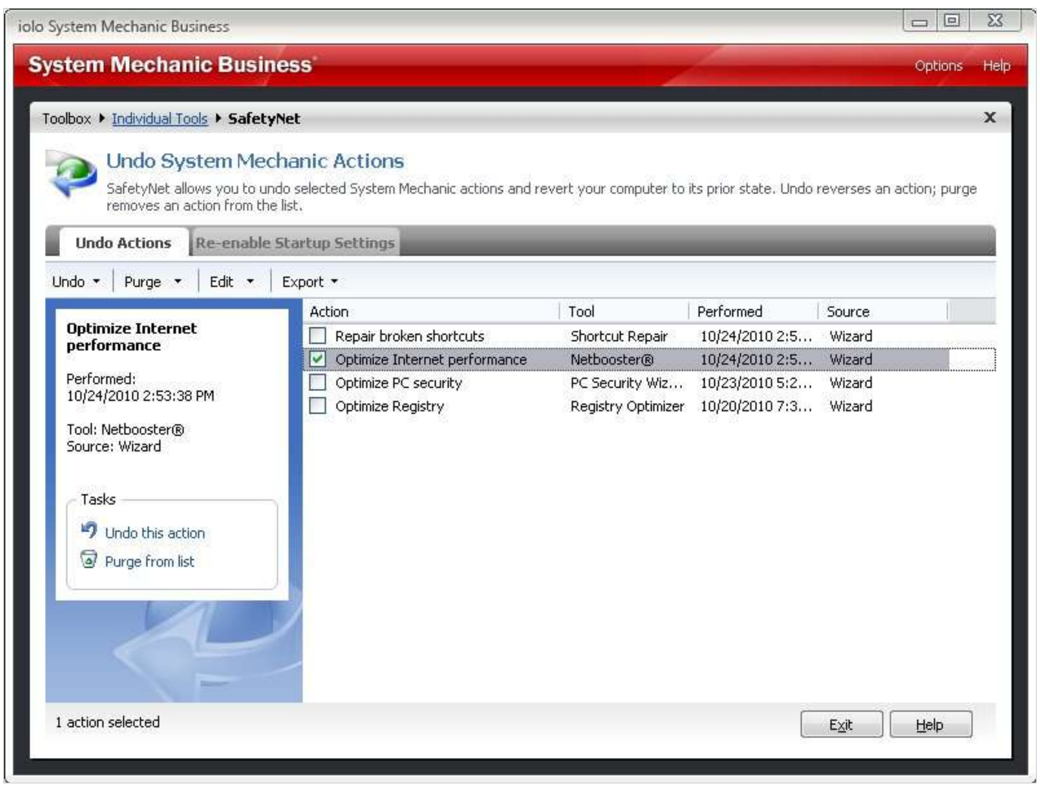
Both of these parameters are fully configurable; they can be adjusted up or down according to the preferences of the administrator. As soon as ActiveCare senses the system is available and not being used, it will perform the specified maintenance and tune-up steps. The instant the user or a running program returns to any significant level of activity, ActiveCare immediately steps back and releases all resources. At the same time, it makes a note of exactly where it left off so that when maintenance actions resume there is no lost time or wasted energy repeating the steps that have been accomplished so far. In this way, ActiveCare is able to cycle through the upkeep of the system over and over again, in the same way that the Golden Gate Bridge is so efficiently maintained.

An additional feature that makes ActiveCare virtually invisible is iolo's ZeroRAM™ technology, by means of which ActiveCare is designed to virtually disappear from memory whenever it is not running. As a result, there is zero impact on your system as a result of leaving ActiveCare turned on. Because ActiveCare performs all critical maintenance actions automatically, without interrupting the user's productivity, and without loading the company's limited IT resources, it completely resolves all of the problems associated with automated workstation maintenance. That's the real magic; it leaves no reason for any workstation to remain improperly configured or in any way not fully optimized.

What Happens If I Make a Mistake?

Workstation users, managers, business owners, and IT professionals, when confronted with a system needing adjustment or repair, all have the same question: "What happens if I make a mistake?" Whether or not it is spoken out loud, it is still sitting there in the back of the mind. Even when the answer is known, such as an IT pro who actually does have a backup plan, the question remains in play. And when there is no backup plan, even the IT pro will hesitate to dig into a system and change its most important settings. People are naturally afraid of making mistakes, especially when it involves large or complex system like car engines, a brain surgery, or the Windows operating system. It seems too risky and the potential for loss, blame, and embarrassment is too high. The upside rewards are often not as visible as they should be in order to counteract the sense of risk. In every business environment, therefore, iolo has consistently observed that whenever this question is asked, you must have a very solid, very clear answer. And in the case of iolo that answer is SafetyNet™.

SafetyNet is iolo's field-proven solution for tracking and managing the changes made to a system through any of the tools in the System Mechanic suite. You might think of it as an extremely detailed revision-control system, tracking each change and providing the ability to undo any of them, either en-masse or one-by-one. The reason the tool is designed this way is simple—with the incredibly large number of settings and interactive complexity of the Windows operating system, even the world's most meticulous administrator could not predict when a particular change might create an unworkable system configuration. The SafetyNet function is therefore essential to obtaining the most efficient management of the workstation, as without it there would be too many negative considerations about modifying the system. An example of the SafetyNet is shown in the screenshot below. In this case the user has made changes to several areas including optimization of Internet performance.



These changes can be quickly and easily undone simply by checking the relevant box and clicking "Undo." Within a few seconds the system is restored to its earlier state. This happens without any major rewrite of the system settings, as would be the case if the Windows System Restore tool were to be used instead. That is, SafetyNet is point-by-point specific, reversing only those changes the user suspects might be relevant, as opposed to wiping out the entire system status and going whole-hog back to some earlier configuration.

To compare the efficiency of each approach, imagine an Advanced Physics classroom with two very large blackboards, each five feet tall and eight feet wide. On the first is an extensive formula describing the construction of the universe. On the second the instructor is copying the formula as he lectures, making various improvements to it as he goes along. At the end of thirty minutes he sees there must be an error on the second board somewhere because the formula now shows the universe could not possibly exist as we know it. This may or may not be a correct result, but it is certainly unacceptable to this one overly excited professor, and so he quickly begins erasing the entire board. Then he starts again, copying once more, line-by-line from the first board, going through the same revision process. Why doesn't he just look at the second board before erasing it to find the one or two places that contain the error and fixing only those instead? Maybe he can't remember the particular changes that he made, and maybe he just doesn't want to spend the time to compare the two boards. In any case by this time some of the students are certainly thinking, "That's one nutty professor." Nevertheless, this is precisely what the Windows System Restore function does on a regular basis. For example, if you have a restore point from one week ago and have since made twenty configuration changes to your system, at least eighteen of which are positive and beneficial changes, your only option under System Restore is to roll the entire system back to the earlier point, in the process wiping out all of those changes. You might as well take an ax to a recently remodeled room and then rebuild it using the old floor plan instead of just patching a couple of holes in the new drywall.

There's another problem with this approach. Now the system administrator has to recreate the eighteen positive changes for which there is no longer any record, or just forget about them entirely. Further, if some of those changes were important and are not being reinstated after the System Restore, he may find himself tracking down several new conflicts in the restored configuration. Pretty soon, then, most people tend to stop using System Restore at all; they go instead to something even more drastic, such as a re-imaging the entire disk.

Compare this to the SafetyNet approach, which takes a more intelligent and surgical approach to the problem. Rather than wiping out the complete collection of system settings, SafetyNet allows the operator to select a small subset of the recent changes and roll them back one-by-one. The operator has full control of the rollback process, rather than abdicating to a deeply buried restoration process. The SafetyNet process is therefore much faster and far more reliable than System Restore, saving a tremendous amount of time, trouble, and headaches. And from the financial point of view, all any business owner or manager needs to ask is, “How do I want my users and IT staff spending their time—solving problems in ways that create new problems, or dealing with them quickly and with finality? ”

The System Mechanic Development Path

The goal of iolo technologies since its founding in 1998 has been to extract every last bit of performance out of every active version of the Windows operating system. A founding premise of iolo’s research and development efforts is the fact that computer operating systems are never perfect. This is especially true when the system is attempting to do more things for more people, while at the same time trying to remain compatible with the never-ending stream of advances in technology—new processor designs and memory architectures, new input and output protocols, larger, faster hard drives, and an avalanche of applications being produced by a world of developers all wanting to do different things in different ways. Therefore it is not surprising, given the monumental difficulty of the task, that the first version of Windows, released in 1985, was not very popular. Version 2.0 did a little better, and then version 3.0, released in 1990, did significantly better. Windows was beginning to seriously take over the market. By the time Windows 95, code-named Chicago, was released in 1995, Microsoft’s share was accelerating rapidly, at the same time as the number of workstations in business environments was exploding. In the year that Windows 98, code-named Memphis, was released, iolo technologies was already hard at work developing tools to fill the many gaps in system stability, efficiency, and performance.

In the twelve years since then, iolo’s original product-development strategy has not changed: Windows operating systems, with the assistance of third-party solutions, still need to be stabilized as the first action, streamlined for efficiency as the second, and enhanced with performance-boosting techniques as the third. The fact that the Windows operating system has become more refined over years has not resolved these problems, as the demands being placed on them have in the same period been increasing at an equal or faster pace. The need for support from specialized tools will continue to be the case until such time as a perfectly stable and continuously self-healing Windows operating system is produced. And there are no plans for that.

The success of iolo technologies in these endeavors can be seen in the rapid growth of the company and in the way iolo's tools have been received by customers in home and business environments and by experts in computer technology fields. From a small handful of staff over a decade ago, there are now several thousand working in 33 countries around the globe, serving more than 33 million users. In both 2009 and 2010, major industry-watcher American Business Awards named iolo technologies the "Fastest-Growing Company in the United States" in the up-to-2500-employees category. In 2009, they also selected Noah T. Rowles, CEO of iolo technologies, as a finalist in the "U.S. Executive of the Year, Computer Software Category."

The hundreds of awards garnered by iolo's System Mechanic suite include Laptop Magazine's "Best Gear for the New Year," the CNET Editor's Choice Award, and a Five Star Rating from BestVistaDownloads, amongst others. One of the most recent is PC Magazine's Editor's Choice, October 2010, which states: "Should You Use Iolo System Mechanic 10? The answer is a resounding yes." While referencing iolo's "excellent PC tune-up capabilities" and "excellent explanation of PC problems," the editors then conclude with the statement that iolo's System Mechanic is "our new PC tune-up utility champ."

In addition to such awards, iolo is especially proud of the many performance benefits being experienced by users on millions of Windows systems, such as those reported by the reviewer below while running System Mechanic on his own workstation:

> *"I saw some very noticeable improvements after running System Mechanic. For instance, boot-up took 1 minute, 57 seconds before I ran System Mechanic, and only 58 seconds after. A restart was 2 minutes, 19 seconds before, and 1 minute 40 seconds after. Time to load a graphics–intensive Web site like CNET was 3.8 seconds before and 2.1 seconds after. Firefox launched in 9 seconds before and 6 seconds after." (New York Times, Gadgetwise Article: "An Easy, Painless Way to Keep a PC in Shape," by Rik Fairlie, September 14, 2010)*

The exceptional performance of System Mechanic in the commercial sector is also being perceived in the public sector, where fast-running Windows workstations are every bit as critical to everyday activities.

"We let System Mechanic do its thing, and within half an hour, our system was humming along like new again. It even scored 20 points higher when we reran the PassMark Performance Test benchmark after the fixes. That's a noticeable and measurable change." (Article: The Great Eight Products of 2009, by Government Computer News, December 11, 2009)

About iolo technologies

iolo technologies, LLC, produces award-winning software that repairs, optimizes, and protects Windows computers. A privately held firm founded in 1998 and headquartered in Los Angeles, iolo is widely recognized as a global leader in the competitive utility software market. As part of its twelve years of operation, iolo has played a pivotal role in establishing the PC tune-up software category, competing with some of the largest consumer software companies to ultimately assert itself as the worldwide leader according to multiple industry authorities. According to market research firm NPD, iolo currently commands 85 percent of the PC tune-up market in the US and 98 percent in Canada, with comparable rankings by similar sources throughout all countries where its products are distributed. Privately held, 100 percent self-funded, debt-free, and profitable since inception, iolo has been one of the fastest growing companies in the Southern California region for more than ten years now, consistently ranking on Deloitte's Fast 50 list of the fastest-growing companies every year since 2004.

The company's product line, distinguished by System Mechanic®, its flagship PC tune-up suite, has become known for its first-to-market feature set and bold initiatives of practical innovation. iolo's vision is to produce powerful, flexible, sophisticated software that is easy to approach and use by both novices and experts, enabling them to enjoy the full potential of their PCs by keeping them running fast, stably, and free of trouble. iolo's patent-pending ActiveCare® technology allows this to occur during times when PCs aren't being used, ensuring maximum productivity without interruption. iolo Labs (http://www.iololabs.com/), the firm's independent research arm, aims to further the company's mission by providing scientific study and insights into the nature of PC performance and how it declines over time.

For further information on iolo and its complete range of workstation maintenance and tune-up tools and related products, see www.iolo.com

7470 North Figueroa Street
Los Angeles, CA 90041 USA
www.iolo.com