

10 Good Reasons to Monitor Your Windows Servers

Introduction

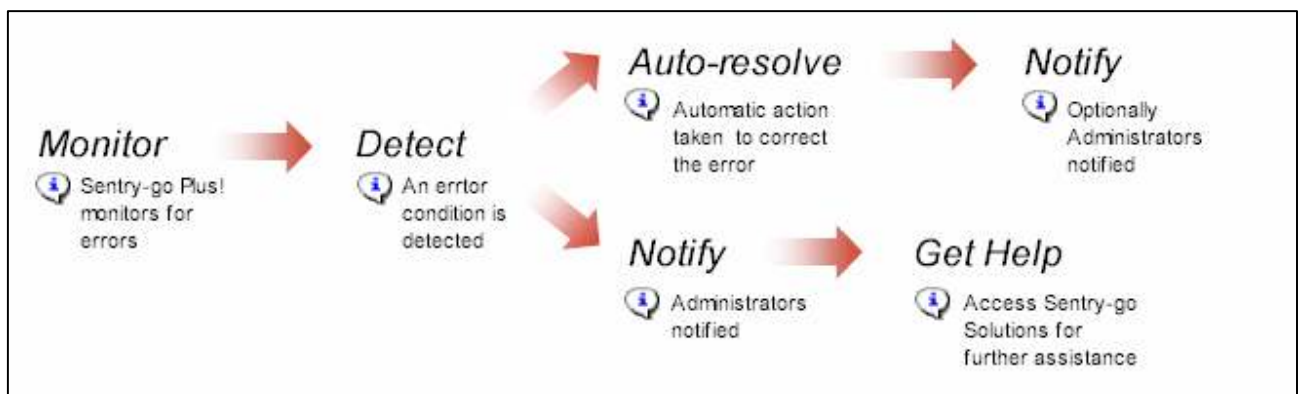
OK, as a company we specialise in automated monitoring solutions so you're already thinking we're going to be a bit biased here right ? Well yes, the first part is certainly true but the aim of this paper isn't directly to sell you one of our solutions – though obviously that would be nice! Primarily, it's to outline some of the main reasons why you would consider such a solution, how it can help, why it is important and why it may make more sense than you've considered in the past.

What is monitoring ?

In essence monitoring is “keeping an eye on something”, the method by which you know your underlying solution is working in such a way that if it was asked to do something, it would probably succeed.

In practical terms of course it's a little more complex. If everything's fine then all a monitor has to do is wake up and check again every so often. But if not, you'll probably want some sort of action – either something it can do itself to put things right or to inform you of impending disaster before that disaster strikes. A good solution should be able to perform all these tasks, and probably more.

The diagram below shows how a typical fault may be detected and handled.



A word about cost

If you're a technician or System Administrator you may only be interested in the technicalities and benefits a solution can bring and we'll certainly come to these shortly. If you're a manager however, you're probably already wondering even if it does sound good, "how much is this all going to cost?" The answer of course isn't quite as simple as a single figure as ultimately it depends on a lot of factors. But what we can say up front is that it certainly doesn't have to cost vast sums of money. It can of course, but it doesn't have to!

As with any project – and when you're setting up monitoring for the first time you might think of monitoring as a project, it costs money. The solutions you can choose from come in all forms too – from a simple shareware application to large-scale systems capable of performing 100s of jobs across your whole enterprise.

With some solutions you'll often pay the price both up front and each year. Costs can include consultancy in setting up the solution and ongoing maintenance. If you want a smaller, yet still extremely flexible & easy to use solution designed to help your support teams then it may be a lot less than you think.

As a guide, the prices of our own solutions, please visit <http://www.Sentry-go.com/pricing.aspx>.

Why bother ? After all, it all seems to be working?!

OK, back to the facts. We know what monitoring is but next there's the age-old question ... "why bother" ? Well, depending on where you're coming from, there are a number of reasons why you ought to at least consider the benefits of monitoring, in particular the benefits automated monitoring could bring for you.

If you've got a perfect system which never fails and works well within its designed capacity then you could save your money and go home! But what if another system uses up all your disk space ? Or what if the network link fails ? These are just two scenarios that might indirectly affect your system even though the system itself remains "perfect".

Of course, most software developers will probably admit their software is, alas rarely perfect ? Its designed to be of course, but things happen that we don't expect – it's asked to do something you hadn't quite thought of on day one, or maybe you did but not in that particular sequence. Maybe you needed to install a Service Pack and now the behaviour of your application is subtly, but definitely altered ?

Lastly of course, is the system actually working as perfectly as you think ? Or have people just worked round all the issues and not bothered to tell you ?

Reason 1. Detect problems within the environment or caused by other systems before they impact heavily on your system.

“Hidden” errors

When Microsoft introduced Windows NT, they introduced a unified way of message logging within applications – the Windows Event Logs. Anyone who’s coded for these will know they’re a bit tricky to start with but in theory at least (and to an extent in practice) they’re a good idea – a set of known locations where you can find details of errors, warnings and information messages. They can also be automatically managed so applications are free to write error details to their hearts content.

However, these logs are a bit like traffic lights in the street. They’re only any good if you look at them and take notice of what they’re saying! It sounds bizarre but how often do you check the logs to make sure no “funnies” have been reported ?

In fact, Event Logs do have a few drawbacks. Firstly not everyone uses them so it’s best to check your applications to see where they log their own log files. Secondly they’re local to the machine – fine if you’re dealing with a single server or workstation but much more problematic if you’ve got a more than a couple to look after. Thirdly, unfortunately a lot of messages tend to get written there – many of which you won’t need to see. The trouble is that these hide the more important ones you ought to be looking for.

Event Logs are one form of “hidden” errors. They’re not deliberately hidden of course, but because they’re not checked and the people using the system don’t see them, they often are. Hidden errors are particularly problematic because it means you go on thinking everything is fine when in reality its not. Users may get unexpected results or in some cases, be blissfully unaware that what they’re doing isn’t going very far at all!

One way of handling this is to monitor your Event Logs regularly but manually checking them in Event Viewer could take ages and you would only be able to review what’s happened long ago. A better way would be to automate this monitoring and let the system highlight when an important error has been written – on any of your servers!

Reason 2. Automated monitoring allows you to be notified of events as they occur without the need to perform manual checks. This includes errors & messages written to both Event Logs & other text-based log files.

Early warnings

It’s amazing what hindsight can tell you. If you review a problem after it’s occurred and been resolved, you’ll often find some early signs that, if noticed, would probably have prevented the problem appearing in the first place. Going back to our Event Logs, a simple example might be in spotting an error the first time it’s written. By understanding and acting on the error when it first appears, the 100s of errors that may subsequently get written – as other users have the same problem, can be prevented.

Another example might be disk space. It sounds simple, but knowing you’re running low on disk space before you run out gives you time to respond. Being told there’s no space left is better than nothing, but by then people and/or systems will have already been affected.

If you’re automating your checks, the system can tell you rather than an irate user or following some midnight investigation as to why your batch process has stopped working.

Reason 3. Knowing about potential issues before they become major problems gives you time to respond and means that other users or systems will continue unaffected.

More obvious errors ... to the user!

Its 8pm and you're logging on to a supermarket's web site for the first time. It's slow to respond or worse still doesn't display the page at all. Do you ...

1. Send feedback to the supermarket by e-mail detailing the problem so they can fix it ?
2. Keep quiet but wait a while so you can try again later ?
3. Go to another supermarket's site that's up and running and shop there ?

In most cases it'll be number 3 – after all, you want your groceries now, not when their sites back up and running! And next time you shop on-line, the chances are you'll go back to the new site because you remember that one was working fine and you don't want to risk wasting another half our trying to shop with the first one again.

This is great if you're the second supermarket, but less good if you're the first. The truth is, most users won't report an error. They'll moan and complain about the system but only at lunch, by the coffee machine or maybe when asked. If only you knew your web server wasn't well you could do something about it but at 3am in the morning who's checking ? Indeed, even at 3pm in the afternoon is anyone making sure its running well, or are you assuming you'll just be told when its not ?

Internal users are often the same. It's not really their fault as they must see 100s of errors a day popping up. These messages often don't mean much to a user ... and anyway, "someone else will have reported it ... and even if I do, nothing will happen so I'll just logoff and try again!" Sound familiar ?

The trouble is, without someone/anyone reporting it, how are we, as IT professionals supposed to know there's a problem ? It may not be an issue with our system, but someone else's may be affecting it in such a way ours stops too.

A better solution is to ensure that your systems are working as the user expects them to and the best way of doing this is to try it for yourself. Logon to the web site, send an FTP file, send an e-mail and make sure it makes a speedy arrival in your inbox – in other words, do what a user would do.

Easy ? Well yes, but rather time consuming – not to mention boring to do, especially as most of the time it'll all work fine! It's well known fact that as humans, we don't monitor things well. We start seeing things we expect to see as opposed to accepting the actual result. We also get tired and start making assumptions – its bound to work 'coz it did for the last 52 days. We then stop checking!

Again, automation here is perfect. Let the system pretend to be a user and perform the check every 10 minutes. Does your web site display a page in a timely fashion ... and does it display the text you expect ? Can you send an e-mail and it arrives in your inbox ? If not, then the problem could just as easily happen to your customer or end user too. And going back to our original example, off they go to your rival supermarket up the road!

Reason 4. By monitoring your systems automatically, you no longer need to rely on end user (or worse still the customer) reporting problems.

It's a 24 hour world

In the old days memos were paper based and people used the telephone to talk rather than type. IT systems also tended to run on large single mainframes or run standalone, operating during normal office hours. Batch jobs might run overnight but they were fairly self-contained and you could always rerun them in the morning if they failed anyway.

Today however, technology has moved on. We now have distributed systems working together across the network, 24 hour access, e-mails that fly round the world whatever time of day or night and fast moving business that means you'll soon be left behind if you don't keep up. Much of this is down to the improvement of IT systems and, to an extent automation. The web server doesn't care what time it is, nor does the database – they're happy to let you order your goods no matter what time it is. But what happens if they fail ? Who will know at 3am or even the next day ? For important systems you will need to know of the fault so checks & repairs can be made; at the very least it would certainly be good to be able to review failures first thing the next day.

If you're not monitoring the chance are you'll never know. A daily trawl through the log files may help but more likely you'll hear about when you're called in to the office to explain why the application didn't work!

Of course, a 24 hour world also includes weekends and holidays. Does your web site for example, operate over Christmas and if it fails on Christmas Eve, when's the next time someone will check in order to get it running again ? After the new year when your office reopens ?

Reason 5. By automating your monitoring, checks can be made day or night, from weekday, through the weekend & holidays. If faults are detected, you can take the appropriate action whenever is most convenient or almost immediately for critical systems.

Who's looking after your business ?

Like other parts of your organisation, IT can be extremely complex – new systems, upgrades, patches, not to mention the admin. of day to day running all have to be factored in. Rather than manually checking once or twice a day when time allows, wouldn't it be better to let your IT staff concentrate on what they're good at ? Designing new & improved systems, looking to the future, seeing how technology can help are all extremely important and very rewarding when it works.

Reason 6. Using automation, IT staff are freed up from the need to perform day to day checking, whilst still being informed if failures do occur. They can then concentrate on problem resolution & improvements as soon as possible – hopefully, before users and customers even notice there's a problem.

Some fixes are easy - some are not!

If your car won't start the chances are it's a relatively easy fix, a fix that can easily be proved by the fact that afterwards the car starts. It can be the same with IT – a bug may be obvious ... “when I do this, the process fails and I get an access violation written to the log”. But other problems can be much more subtle – maybe intermittently failing, maybe only failing when other conditions also occur.

Reason 7. With full & accurate details of when a fault occurred – maybe even as it occurs, you can more easily spot trends or perform diagnostics at the right time, thus helping pinpoint failures or forward them to third party suppliers etc.

Systems evolve – even successful ones!

In an ideal world, new systems will be designed and left to run for years. But things rarely stand still and it's very likely that they will be revised and upgraded over that time. A successful system is just as likely to be enhanced as it becomes more popular.

In addition to functionality updates, not to mention fixes and patches, there's also the issue of performance. A system designed for 30 concurrent users may be fine for that number - or even 40. But what happens when 100 users log on ? Or when the system's database is accessed by a new web interface you've just paid a lot of money for ?

Of course, it's not just number using the system that's important, but also the affect they have on other parts of the system too. Take a database for example –your “orders” table has a trigger on it that's fired each time a new order is placed. If one or two users order goods at a time then that's fine. But what if your on-line store's a success and many more users begin to hit it ? And what if a later change alters things such that the first trigger causes others to be fired ? Before you know it, your “perfectly good” database design has hit a snag, it works in theory but all of a sudden it's grinding to a halt with lock contentions and all sorts.

In this example, by monitoring both throughput and concurrency, you could easily profile how the application or server is being used, where the bottlenecks are and also highlight when you're approaching the maximum designed limits. If you're increasingly running at capacity it would be much better to know sooner so you can upgrade the hardware or software etc. rather than wait for the system to fall on its knees.

Databases too are the key to many systems. It doesn't matter how nice the interface is, if the database can't handle the throughput – or the design doesn't let it, you're in trouble. By monitoring locking within SQL Server you can easily highlight both isolated and recurring problems, allowing you to act quickly and before the issue gets way out of hand.

Reason 8. Using automated monitoring, you can continually monitor the performance of your key applications and be warned when design limits or thresholds are exceeded or conflicts arise.

Help the business

Not all monitoring is about spotting errors. You may simply want to automate a particular task to help your business. For example, being notified when a new file is uploaded so you can access it prior to forwarding to another group. It may be you simply want to know when files are updated so you know to update your web sites or other parts of the system. Equally, you may want to know when the number of rows in a database table exceeds a certain number or when the results of a query go outside permitted limits.

Reason 9. Monitoring solutions may be able to help streamline the IT processes within your business.

Automatic Responses

In general, a monitoring solution is designed primarily to monitor one or more aspects of your server and alert you when an error condition is detected. These alerts may take the form of a network message, e-mail, SMS text message or messaging via a 3rd party solution.

However, there may be times when the immediate action is more obvious. For example, if a service fails the first thing you would probably do is try to restart it. If you're running low on disk space, you'd clear down temporary files and check to see how much space remains after that. If you have a locking problem between two processes on a database, you'd terminate one to allow the other to proceed. If the monitor can do this for you then so much the better. That way you'll be informed of the failure and can concentrate on the cause safe in the knowledge that immediate action has at least been taken.

Reason 10. If your solution can be configured to handle known issues itself, you are free to look into the cause whilst the impact on the end user & customers is minimised.

OK, but what's the overhead ?

"Its all very well continually monitoring the system, but what impact will the monitor itself cause on both the server and the systems its checking ?"

We get asked this question quite a lot. Obviously nothing is free and any process run on the server will take up a certain amount of resources. However, its important to note that the word "continually" is a bit inaccurate – most monitoring is performed periodically not every second or two. And the overhead itself is dependent (i) on the tools you are using and (ii) what you're asking them to do.

But in essence, a good monitoring solution will be unobtrusive, working in the background and utilising only the resources it needs to run. This can be achieved in a number of ways – one example, with many of our own solutions is to run the monitor locally wherever possible. This way they don't impact (or rely) on the network except for alerting and if that fails, another monitor can sound the alarm.

“But what should I monitor ?”

This is another question we often get asked and the simplest answer is anything that is of use to you – in other words anything you would act upon should the condition be reported to you.

There's no point being alerted to, say, the number of Microsoft Exchange users logged on if (i) you don't look after Exchange and/or (ii) you have no idea what the system is designed to support.

However, if you're the DBA and want to monitor SQL Server locking, your database sizes, the SQL Server log file or ensure the SQL server service is running, then that's what you should be able to do.

Other more obvious checks might include ...

- Available disk space
- Ensuring “auto-start” services are running
- Important software processes are running – such as anti-virus software
- Errors written to the Event Logs or log files we spoke about earlier
- The ability to access web sites or e-mail systems by mirroring what a user would do

“Won't I just be inundated with faults ?”

Err ... firstly this is an actual quote from a client (!) and secondly how are you supposed to answer it ?! In fairness, to an extent he had a point, but only because he knew deep down there were problems. If you follow the theory that “what you don't know can't hurt you” then remember that it may not be hurting you but it may be hurting your business, not to mention your reputation! Eventually someone higher up will probably notice and want to know what you're doing about it. Saying you don't have any facts (which therefore means you're doing nothing) probably isn't going to wash for long either.

The answer is simply to find out. Just because a fault is reported doesn't mean you have to act there and then – depending on what it is. You simply work on the important ones and keep a check on the others. As your systems improve (because you're now able to highlight and fix problems) you will have time to look at the other issues as they arise. Remember too, that some faults won't be yours, they'll be in or potentially linked to third party products – being able to provide hard facts as opposed to simply theories will help the other company's technicians resolve the problem more quickly too.

Of course the up-side of this means that you can also monitor systems as improvements are made.

Choosing a solution

Having highlighted a number of reasons as to why you should consider an automated monitoring solution, which product should you choose ?

Well just because we're writing the paper, doesn't mean we're going to start up a whole sales pitch here, though naturally we'd like you to consider our solutions, detailed briefly below as well! The best advice is to think about what you want to monitor and purchase a tool that meets those requirements, is easy to configure, is within your budget and is something you think you or your team will be comfortable with. Some vendors, like ourselves allow you to download & run an evaluation free of charge beforehand – an ideal way to see the benefits but also confirm that the solution will both work and help within your environment.

Also, as a guide you may also want to consider the following ...

- How the system works and its impact on your technical environment, particularly the network
- Does the way it alerts fit in with your work practices ?
- Is it flexible enough to expand as your environment changes ?
- Do you pay per user, per server or per check ?
- How good is the after-sales support ? This is more difficult to check but maybe discuss your requirements with them via e-mail during the evaluation to see how they respond. Also, some vendors offer advice, maybe through their web site in order to help you resolve problems detected by the monitor which can be extremely useful.

Sentry-go® Monitoring – *Be Proactive, Not Reactive!*



As specialists in automated monitoring solutions for the Windows platform, we develop and support a complete software range.

Called Sentry-go, it is designed to provide quick & easy solutions at a very affordable price whilst allowing you to purchase the elements you need rather than features you can't or don't wish to use.

Features of Sentry-go include ...

- Plug & play monitoring features – you only purchase the components you need for your server
- Out of the box monitoring for one or more dedicated tasks
- Flexible licence options ...
 - Quick Monitor for pre-configured monitoring of standard software environments including SQL Server, IIS, Exchange Files & Directories etc.
 - Individual Sentry-go Plus! licences for specific/targeted monitoring
 - Combine licences to meet your server's monitoring needs
- Fully customizable monitoring with the scripted monitoring component
- Automatic responses for known faults
- Alerting options inform you of detected failures, including e-mail, network message, SMS or 3rd party applications.
- Integrated web server provides dynamic web reporting direct from the monitor
- Logging to CSV and ODBC database such as MS Access or SQL Server.
- Quick to install, easy to configure.
- Central console-based monitoring of alerts & configuration
- Access to the Sentry-go Solutions web site for help & advice on resolving detected problems.

For more information, contact us today or visit <http://www.Sentry-go.com>.



About 3Ds (UK) Limited

3Ds (UK) Limited is a privately owned UK-based software house established in 1998 which specialises in the Microsoft development platform and in particular monitoring software. We develop & support a full range of monitoring solutions which provide our customers worldwide with affordable, automated monitoring solutions that are both quick & easy to use, whilst providing support teams & management with the information they need access to.

We also provide consulting services as well as bespoke software development & training to our clients. For more details please visit <http://www.3Ds.co.uk>.

Contacting Us

If you would like further information on monitoring, or any of the support or development services or solutions we offer, however large or small, we'd be happy to talk to you.

- | | |
|-------------|---|
| ✓ E-mail | Contact@3Ds.co.uk
<i>We aim to respond to all e-mail enquiries within one working day</i> |
| ✓ Post | 3Ds (UK) Limited,
69, Esher Road,
East Molesey,
Surrey.
KT8 0AQ
United Kingdom. |
| ✓ Telephone | (+44) (0) 1932 225349 |
| ✓ Fax | (+44) (0) 208 399 4923 |



3Ds (UK) Limited
Design, Develop, Deliver Solutions!

69, Esher Road,
East Molesey,
Surrey.
KT8 0AQ
<http://www.3Ds.co.uk>