Website Availability -

Maximizing Website Uptime

A White Paper from

MyWebAlert.com

(WebMonitor version 1.0)

June 2005

http://www.mywebalert.com

Contents

1 Introduction – The Problem

This White Paper is intended for anyone who thinks their website has some value and is worth looking after. Fair enough, that is a pretty pejorative statement. Everybody thinks their website has value or else why was it developed. Think over the last few years and figure out how many hours you have spent agonizing over the content of the site, getting the graphics just so, sharpening up the download times, promoting on search engines, tweaking adwords, introducing a shopcart, maintaining the content … the list goes on and on. And, of course, it is not just hours spent. More than likely it is also dollars spent.

Now answer honestly - do you know if your website is up and running properly as you sit reading this? For too many people the answer is no. Sometimes they kid themselves that they or someone else in their organisation checks every now and then or that their customers will tell them that they cannot get to it. Even where this is really the case, can you be sure that an internal check means that external customers can get to the site or that a customer report is actually a customer problem rather than a website problem?

OK, let's assume that your website is down. Do you know what to do next? How do you identify the problem and sort it out?

This White Paper tackles the two questions:

- Is your website running; and
- How to fix it if it's not

by providing a few tips and tools – we hope it helps! Even if you have heard most of it before, we would like to think that the **checklists and tools** provided help you remember what to start looking for when the inevitable happens.

2 What can go wrong

You would probably be shocked to know just how common it is for websites to be "down" in some sense. Browsing the Internet there are so many broken/out-of-date links, network problems etc. that you just don't notice. It becomes habitual to maybe retry once before cursing the owner and moving on to an alternative, so much so that you barely notice it happening. Of course the same is true for your customers.

Our survey of local government sites in Britain, where all services are now meant to be web delivered, found that the mean-time-between-failures (MTBF) for the worst sites was as little as a few days.

Broadly speaking there are 4 types of thing that can go wrong – user, network, site administration and site technical.

If a user has a problem and does report it to you, the first thing you need to know is whether the problem is with them or you, and the best way to determine that is to have your website monitored automatically, so that you don't have to investigate at the time of the report. If you know the problem is not with you then you can productively invest some time helping the user – may be it is their security settings, bookmarks, etc. Whilst if you know the problem is with you, you should already be working on a solution, and can give the user some useful information indicating that you are on top of it.

Similarly, when there is an issue with the intervening network, although there is often not much you can do about it, if you have the appropriate automated monitoring in place you will know where the problem lies and will not waste time investigating your own or your user's technical and administrative setup.

This white paper focuses on the technical and administrative types of failure that can beset your website, for example:

- You forgot to renew your domain name or hosting contract!;
- DNS mis-configuration or un-notified change;
- Credit card/Payment Interface issues;
- Hackers have corrupted your pages;
- Server overloading;
- Database not working
- FTP not working;
- Email not working.

In the following section(s) we indicate how this can happen and how you can recognize it.

3 Manual Investigation

OK – there is something amiss with the website – what do you do now?

3.1 Inventory

Those of you who have been around IT for long enough will know that one of the best tools in a disaster is a document telling you what a healthy system looks like. Whilst data and applications can be backed up, it is not always easy to back up configurations, and although it might be in your mind now, in six months time when the failure occurs, will you remember, and if you are not around, will anyone else remember? Also, a good rule of thumb is: the most likely cause of failure is the last thing that changed.

So first off – make life easier for yourself and do a bit of housekeeping **NOW – before the problem strikes.** Let's offend the security purists first! Record all the accounts, usernames, addresses, **passwords**[1] and directories, including backups (you do those regularly don't you?) where your website material is. As well as the obvious, i.e. your web hosting account, don't forget any other providers such as where you bought your domain names etc. Sure, put them under lock and key, but if you can't get in or contact the provider, there is absolutely nothing you can do.

Next, record what the working system looks like. On an individual level, failures may only occur every year or two and it is very easy to forget, so write it down. It could be something as simple as my server's IP address is 1.2.3.4 or this error logfile or mailbox should be empty. This makes it much easier to spot what has gone wrong. See the sections below for suggestions on what is worth recording.

When the site was set up you almost certainly went through some selection process to ensure that it had the right facilities, e.g. versions of Linux, Windows/NT, Apache, IIS, PHP, SSL/SSH, Perl, MySQL as well as capacity e.g. disk space, bandwidth, number of email accounts or ftp accounts, etc. Make sure you recorded what they were when you bought it because most ISP's don't bother to warn you when they change this stuff. In fact the support staff often don't know exactly what has changed. We recently experienced a spate of obscure failures on a couple of our websites. One of the ISP's swore blind that they had "changed nothing that would affect our code". It turned out they had upgraded Cpanel, which in turn had included a new version of the CPAN Perl DBD::mysql

---

[1] Passwords are tricky. We recommend that they are listed separately and kept separately from the rest of the account information to which they relate, and that you link them by some form of "password memory jogger".

library. Since we had a record of which library version worked and the new version library had only been released the day the failures began to occur we were able to prove what was causing the problem and get them to revert to a working system.

Our inventory spreadsheet provides a template for the sort of record we suggest you keep. Note, you really do need to go down to the level of individual library versions for everything you use. The second point is you will probably start off with the good intention of keeping it up to date, but since this stuff changes so rapidly you would have to do it at least every week and that soon gets tedious. The only solution is to automate the recording of inventory values keeping a historical record of what changed & when it changed. It's not difficult to write a little bit of perl, put it on cron then post process the results into csv or whatever, do a diff & generate an alert whenever something has changed. With luck you will get the alert before it all starts falling apart on you! Also, if it does start falling apart it is much quicker and easier to report the bug and expect to get it fixed if you can supply full information about the total environment.

We have included a simple perl script as an example of how to test the perl library versions in use.

3.2 DNS

The DNS (Domain Name Service) translates the human-readable site name, e.g. http://www.mywebalert.com, to its IP address – something almost completely unlike 255.255.255.255. Since no one with a personality knows the IP address of websites, if the DNS doesn't work for your site then your site **IS** down - as far as customers are concerned anyway.

The kind of error you get when trying to browse to your site will be something like the typical IE error "Cannot find Server", or if you can see an HTTP 400, 500 or 504 error code in the website logs.

So whose problem is it?

If you master and manage your own DNS then kick your system administrator – chances are he edited something recently and screwed up.

If you got the Domain Name with your hosting package from your ISP, then contact them, show them the logs, ask them if they moved your site between servers, i.e. different IP address, or if they have been modifying their DNS servers.

If you bought the domain name separately from the ISP hosting package, then the first thing to check is the http/domain forwarding setup by going to the administration page of the name service provider – you did put those details in the inventory and you have renewed the subscription haven't you?

A quick check can be made with the MWA_DNS.pl script supplied with this document. This uses the Net::DNS perl library to find your DNS servers then queries each in turn with dig (Domain Internet Groper) for the Resource Records they hold.

Alternatively, for a Windows server you could use the netdiag tool:

```
netdiag /test:DNS
```

(This tool is freely downloadable from Microsoft at http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/netdiag-o.asp).

Don't forget there will be a master and one or more secondary DNS servers for your domain name and that the DNS caches results – hence you can quite often get intermittent, delayed or ambiguous results as one DNS server may return valid answers whilst another does not.

3.3 IP Routing

If the Internet can translate your website/domain name into an address, its next problem is to make sure information can be routed there and back. IP routing is a real black art. In fact it is a wonder it works at all since it relies on a load of techie things collaborating.

Unfortunately the tools to check that this stuff is working OK tend to be restricted as misuse can overload the Internet.

The old favourite is, of course, **ping** which tries to send an ICMP (ECHO_REQUEST) packet to your host and get a reply back. Most ISP's won't give you access to it via the shell, but if you are lucky enough, don't abuse the privilege. First, as part of the inventory, you should record the routing path into and out of your website.

The first step is to use **ifconfig** to determine if the local interfaces are up. In particular check the local Ethernet interface: eth0, and the Loopback interface lo, are not reporting errors, dropped packets or overruns.

Next, check the local IP routing to ensure packets are routed via the local Ethernet interface with the **route** command.

netstat will show you the current network connections but it can be difficult to interpret and, anyway, most ISP's won't give you access to it.

To avoid unnecessary traffic, overloading the system and having the chief system administrator barring you for life, you should always ping, one hop at a time, out from your server.

The commands listed above can be run directly from the Linux command line. On Windows systems, most of them have variants that are available as options to the **netdiag** utility mentioned above.

Ping tells you if packets are getting lost and the delays they experience – handy if the problem is intermittent. A more sophisticated tool is **traceroute**, but this imposes even heavier loads looking for ICMP (TIME_EXCEEDED) responses. Unfortunately it has been so over-used that almost no ISP's will give you access to it. The only alternative is to try it manually from the few service providers who allow it, e.g. the lookingglass servers:

http://www.nanog.org/lookingglass.html

Or registering with Visualware who provide a free service with a neat graphical interface showing you where your packets go and where they get lost:

http://visualroute.visualware.com/

If the problem looks like it's close to your ISP, e.g. the default gateway, the switches connecting them onto one of the backbone carrier networks etc., then they should be able to fix it. If it is further out in the Internet, then telling them what you can see, as soon as possible, at least gives them a fighting chance in dealing with the next network operator in the chain.

3.4 Server Overload

If the symptom you are getting is intermittent timeouts, e.g. HTTP 500's, it could well be that the server is experiencing temporary overload – quite common on cheaper shared (virtual) servers.

On Linux systems, the most useful tools for checking the (lack of) system resources are:

df	to check the available disk space – you haven't just blown your allocation have you?
ps –ax	to check what processes are running – you haven't been creating zombies or spawning more children than have died have you?
top –s	to check the most CPU intensive tasks of the system
free	to check the free and used memory
vmstat	to check virtual memory statistics, swap space, io and CPU idle

On Windows systems **Task Manager** provides similar information.

3.5 Server Services

Obviously we don't want to regurgitate all of the books on Linux and Microsoft system administration, but you might find some of the tips listed below helpful – they are common gotchas that we confess have got us in the past.

First try browsing to your website yourself!

Next, if you are using an ISP rather than hosting your own service, remember to take a quick look at their opinion of the status of the server hosting your website, this status is usually available from their website not your control panel, and try connecting to your control panel. Have both of these links as favourites in your browser and keep a record of them in your inventory.

Most ISPs give you access to raw server logs and a synopsis of errors. Remember to check these early in your investigations. Formats of these vary somewhat. For help in interpreting them see:

http://publib.boulder.ibm.com/tividd/td/ITWSA/ITWSA_info45/en_US/HTML/guide/c-logs.html#common

Do you use automated scheduling of your application process and is it still working? Check the cron process and your crontab records on Unix servers or your Scheduled Tasks and Services on a Windows server.

If your website depends on file transfer for any functionality then you should test that that is working by trying a download or upload. It is a good idea to set up some form of config file / bookmark / shortcut with appropriate parameters so that you do not have to remember and type correctly when the time comes for a test.

Do all of the files on your website have the correct permissions? Sometimes permissions can get accidentally changed if a website is reloaded from backup or when pages are uploaded following modification. So if your website requires some files to have permissions that are different to the default (for example they need execute permission), you may find this is the problem. You should keep a record in the inventory of any files for which this is the case.

Have files been uploaded in the wrong mode? Shipping binary files as text or text files as binary can have some very interesting effects, usually detrimental, on website functionality.

If your website is commercial, chances are it has a database behind it, and without the database the site is pretty useless. As there are so many different types of database you could be using, all we can suggest here is that you use your standard database administration tools to check its general health, consistency, performance and size on a regular basis. Many ISPs (sensibly) prevent users from gaining access to database administration tools and in this case it is a good idea to write a simple test probe of your own that you can run if you suspect a database problem. The simpler the better because you want to be sure it is the database you are testing and not some complex interaction with other things. Perhaps the simplest is just to try and gain access via your control panel (if this feature is offered), and run something that shouldn't ever fail like 'SELECT NOW();' or 'SELECT 1;' or 'SELECT * FROM <table>;' where <table> is one of the tables in your database..

3.6 Interpreting Error Codes

If you have done all this investigation then you probably have a whole bunch of diagnostic information that includes error codes from various protocols. The following list of links might help to interpret them further.

If you simply want to check the HTTP return and error codes then see:

http://www.mywebalert.com/rfc/2616.htm

If there are any DNS failures and you want to check the meanings of those return codes see:

http://www.mywebalert.com/rfc/1035.htm

If you run Apache, then try their FAQ which is pretty good:

http://httpd.apache.org/docs/misc/FAQ.html

Whilst if you run IIS, your best bet is obviously the Microsoft support and knowledge base which has decent search facilities and lots of articles on specific problems:

http://support.microsoft.com/

Finally there is Microsoft's interpretation of HTTP error codes at:

http://support.microsoft.com/default.aspx?scid=kb;en-us;173971

4 Helping Your ISP To Help You

First, record how to contact your ISP for support in your inventory document, especially if account details are required. Again the need to contact your ISP happens so infrequently that it is easy to forget, and it is sometimes difficult to find out, how to do it. Also note any peculiarities of the system. Some ISP support desks do not proactively inform you of changes in a ticket's status, asking you for further information, for example. The last thing you want is for both you and the support desk to be sitting there, each waiting for the unsuspecting other to do something.

When you raise a ticket you should tell them everything that does work and everything that does not so they can work out any correlations. Try to stick to the observed facts and make it clear if you are making any inferences beyond these, and send them all the supporting data you can.

Say when you know for sure that it last worked, and when you first noticed it was not working.

5 Automated Monitoring

Once you have got to grips with checking and debugging your site manually, the next obvious thing is to apply a little automation, to give yourself a head start by knowing if anything is wrong as soon as possible.

5.1 DIY Solutions

The commands listed above are easy enough to piece together into shell or perl scripts and run automatically to email you either when things go wrong or, with a little more imagination, regularly with a status report warning you that they are about to go pear-shaped. This is fine if you have the time and inclination or are very strapped for cash.

The main issue is where to run your solution from. Obviously it is not a good idea to locate the monitoring software at the site being monitored. Equally obviously the solution needs to run 24/7 and so there is not much choice but to invest in another, distinct ISP account.

5.2 Proprietary Tools

The next level of automation is to buy one or more of the proprietary application, website, server and/or network monitoring tools such as Alchemy Eye, Network Monitor etc. These are pretty cheap, around $2-400, relatively easy to configure – say about a day to configure and tune it to get a sensible balance between false alarms and never knowing if the site is OK.

On the whole this sort of solution works pretty well, with the major exception, that you can never be sure whether it is the monitored site, the monitoring software or the intervening network that is the problem. To overcome this, you need monitoring from several independent locations and some correlation of the results, which brings us onto the final level of sophistication - hosted services.

5.3 Hosted Services

The real beauty of these is that there is nothing to install, configure or look after. You just point them at your website and they do the rest. Things to look out for include:

- How many separate locations they monitor your site from? One is a waste of space – is it down or are you? Two is not much better because between them then cannot know which is telling the truth. So the correct answer is a minimum of three;
- Do they give reliable results or false alarms? You will soon learn to ignore a cheap service that cries wolf too often and if you ignore the alerts, what is the point?
- How often do they monitor your site? Here is no right answer here. The more often they visit your site the sooner you will know when it's down. However, as it can be a resource-intensive activity, the more frequently it polls the more it is going to cost;
- Do they give you any diagnostics? It's good to know as soon as your site is in trouble, but much better to have a helping hand in knowing where to look to get it back up fast;
- Do they check the website content? A corrupted website is often worse than no website at all.

6 Inventory Scenario

As a companion to this white paper we have included an inventory spreadsheet that we strongly recommend you complete for future reference. The spreadsheet supplied has been completed for an imaginary company, Acme Inc, by way of helping you to understand how to use the spreadsheet (although it has to be said that it is pretty straightforward). You should not be afraid to add/remove rows and/or sheets as you tailor it to your environment. The important thing is to complete it and keep it safe.

It may be of some further help to give a brief description of Acme Inc.

Acme Inc manufacture green widgets and sell them exclusively over the web. They bought their domain name, www.green-widgets.com, sometime ago from a supplier who still hosts the domain name on its DNS servers, but the website itself is now hosted on Christmas Island by Christmas Hosters. They also use the Bank of Christmas for taking online payments. The website server runs the Linux operating system; the shopcart and payment interface are written in Perl and use the MySQL database. In fact two databases are run by Acme, one for the production service and one for testing. Of course, their website is monitored by MyWebAlert!

This is a fairly typical scenario, which is relatively simple. Even so, it is instructive to note just how much information may be required when the website goes down and to consider whether you would easily be able to get it without the spreadsheet.

7 Postscript

If you want to send us any feedback, e.g. to add any tools and techniques, or even to disagree, please email us on:

info@mywebalert.com

If you want to roll your own – best of luck.

Alternatively, if you would like a service that takes most of this hassle off your hands for a few dollars a month, have a look at:

http://www.mywebalert.com