Whistle Blower 2.0

Whistle Blower is a server monitoring utility. It can regularly connect to your servers and verify that they are responding properly and within a reasonable amount of time.

If a problem is discovered Whistle Blower can alert you via e-mail, send a page via either Simple Pager or MarkSpace's Page NOW!

If you have a Master Switch from APC or a Power Key from Sophisticated Circuits Whistle Blower can actually restart a failing server without any further intervention from you. Whistle Blower also fully supports the new KickOFF! server restart device from Sophisticated Circuits.

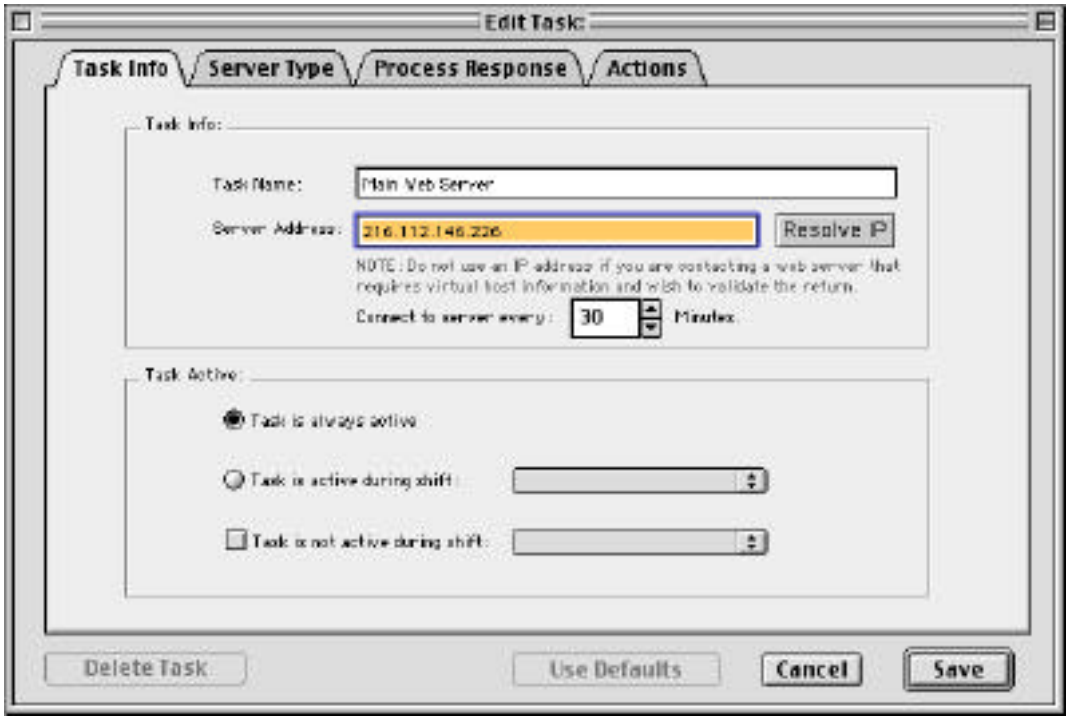
Whistle Blower can also launch Apple Scripts to help you customize your response to an outage as well as play custom sounds or speak phrases as part of the alert process.

Whistle Blower can directly connect to the following server types:

- Web Servers
- Apple Share IP
- DNS
- Filemaker Pro
- First Class
- 4D Servers (4D Open Required)
- FTP
- Hotline
- LDAP
- News
- POP3
- QuickTime Streaming
- Radius Authentication
- SMTP
- Telnet
- IMAP

You can also verify a server by sending a standard Ping and connect to anything that listens for incoming TCP connections via the Generic TCP task.

1. Setting up a task



The first time you run the program the Edit Task window will come up. At any other time you can select 'New Task' from the File menu. Double clicking on a task in the main window opens the Edit Task window for editing the existing task.

Task Name: Every task must have a unique name that will be displayed in the Main Window display.

Server Address: This is the DNS name or IP address of the server you want to connect to. By entering the IP address, or entering the DNS name and clicking on the 'Resolve IP' button (PPC only) you will still be able to connect to the server if your DNS server should go down.

1. Setting up a task

Connect To Server Every: How often you want to verify the server.

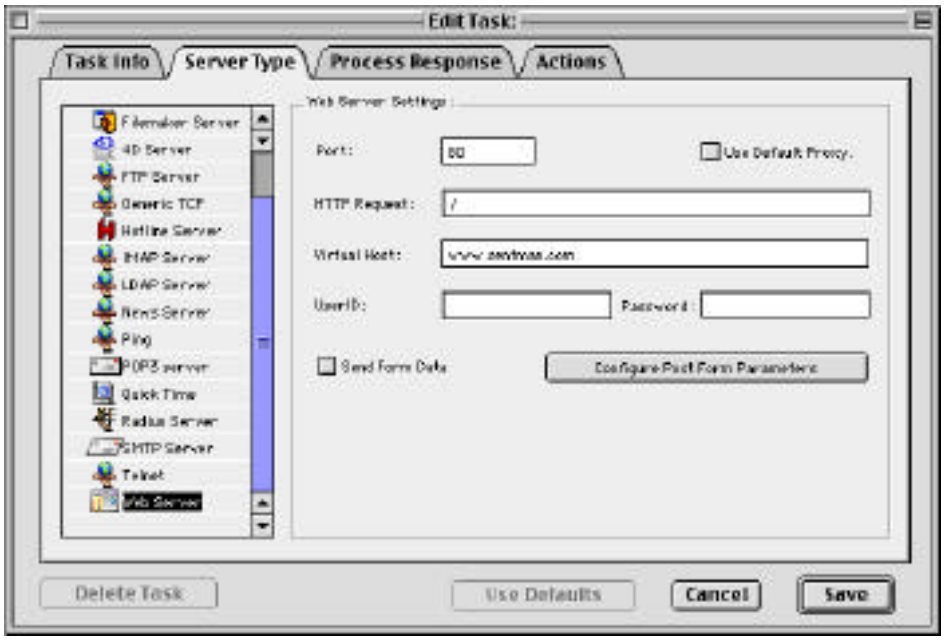
Task is always active: The task will check every interval.

Task is active during shift: Shifts are discussed in detail later in this documentation. If you select this option and a shift from the popup dialog the task will only check while the selected shift is active. This can be useful if you need to have certain checks run only during certain times or certain days.

Task is NOT active during shift: If you have regular backup times, reboot times, or other downtime on a schedule you can use this setting to keep the servers from being checked when they are supposed to be unavailable.

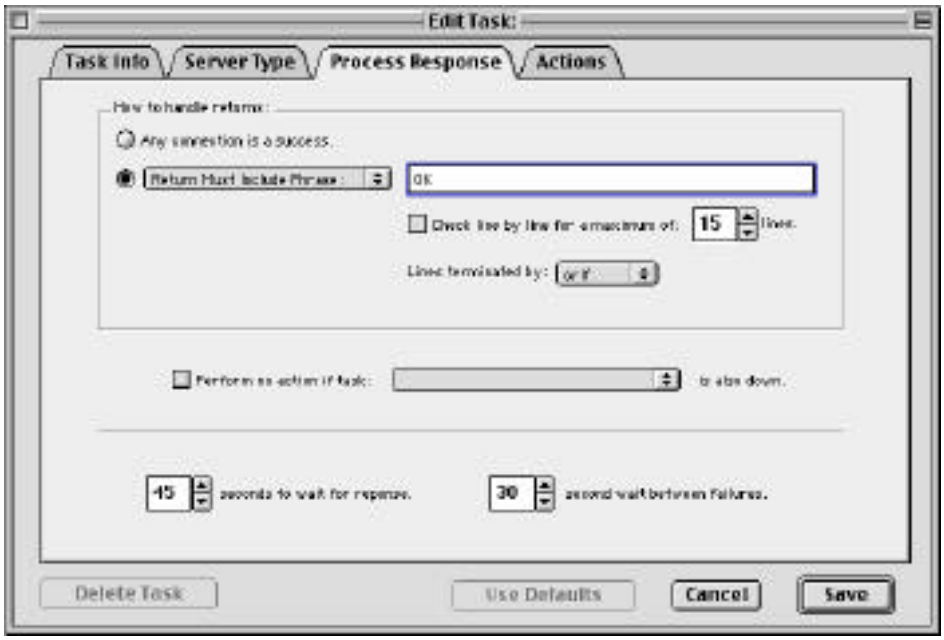
Use Defaults Button: Once you have a task created you can select it in the main window and select 'Make Default' from the task menu. If you click the Use Defaults button it will fill in all the information from your default as a starting point for you.

1.1 Setting up a task *(Server Type)*



Server Type: Select the type of server that you would like to connect to from the list at the left. Any other settings that are necessary will be displayed on the right hand side after you make your selection. A more detailed discussion on the setup of the individual server types is available in the Server Types chapter.

1.2 Setting up a task *(response)*



Any connection is a success: The exact functionality of these settings is dependent on the specific server type selected. Generally selecting this means that no conversation is made with the server. If it opens a connection properly then it is considered a success. The exceptions to this are servers that would log an error if the connection were just terminated. Web servers wait for the page to load and SMTP and POP3 servers wait for the HELO message and then send a proper QUIT message.

Return must/must not include phrase: allows you to validate the response from web servers and generic TCP streams. For example, if you have a web site problem, the server might still return an error message page. So if you had selected the 'Any connection is a success' it would still think the server was up. If you enter a phrase that you know is contained in the page you can trap for this situation. In combination with the 'send post parameters' on the previous tab you can verify that web forms and databases are returning the proper responses and not error messages.

1.2 Setting up a task *(response)*

Check Line by Line: This is not a setting that you will have to use under most circumstances. If the web server forces a 'stay open' type connection and also fails to give a content-length header then WB wont know when it's gotten the full page. So it will time out unless you tell it to only look through the first n lines for your phrase to verify.

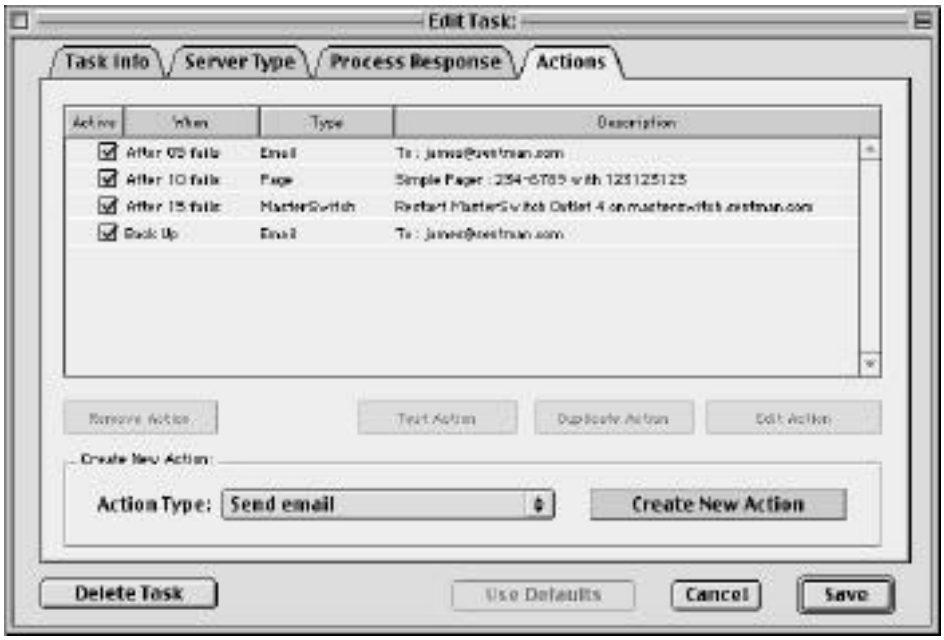
This also makes it possible to verify the return from an open ended TCP stream using the Generic TCP method.

Perform no action if: If you select a task in this popup, then before performing an action on failure Whistle Blower will run the dependent task and make sure that isn't already down. For example, it makes no sense to be restarting the web server if the router is down.

Seconds to wait for response: This is how long Whistle Blower will wait for the server to respond. If you are checking a web server this is how much time the program will wait for the whole page to arrive. So you may need to adjust this if you are on a slow network, if the web page is very busy, or if Whistle Blower is running on a very busy or slow Mac it may take longer to receive the information. For most kind of connections the 45 second default works very well.

Seconds wait between failures: You may be connecting to your server every 30 minutes, but if a connection fails you don't want to wait another 30 minutes to see if it's come back up or to perform an action. Once the first connection to the server fails Whistle Blower will switch to this shorter interval for checking so that you can better gage your response to the server.

1.3 Setting up a task *(actions)*



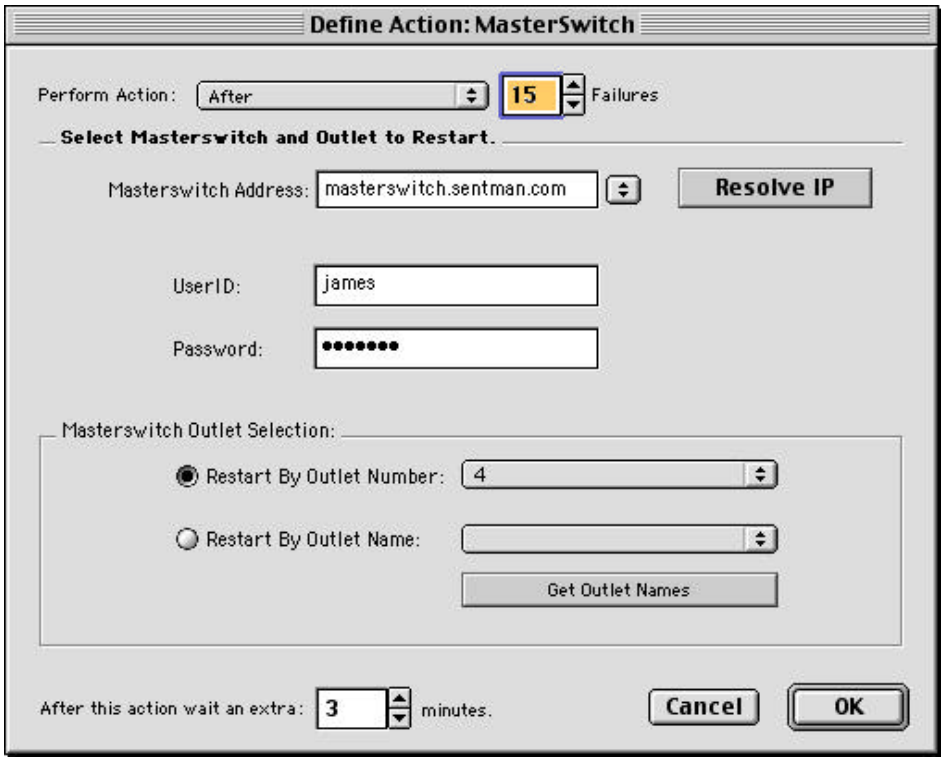
The new action system in version 2.0 allows you to setup your response to an outage based on how long the server has been down. Many servers will occasionally refuse a connection and this does not necessarily mean that the server is down. In the example above I've setup my task to ignore anything less than 5 failures in a row. At 5 failures it will email me. If I haven't fixed the problem by 10 failures it sends me a page. If the server is still refusing connections after 15 it will restart the outlet on the Masterswitch that it is plugged into. Then I added another action to send me another email when the server comes back up.

You can disable individual actions via the check boxes in the 'Active' row of the actions.

The edit/Duplicate/Remove buttons should be self explanatory. To test an action highlight the action and click the test button. Be careful though, if the action is to restart a powerkey with a live server attached to it, testing the action will actually cause the server to be restarted.

1.3 Setting up a task *(actions)*

Creating a new action: Select the type of action you wish to add from the popup menu and click the 'New Action' button. This will open the Action configuration window containing the controls necessary to configure that type of action. Here is the window for restarting a MasterSwitch.



Perform Action: All action types have the 'Perform Action:' selection at the top. You can set the action to happen after a certain number of failures, as in the example. If you would like the action to continue to happen as long as the server is down, for example keep sending you emails until the server comes back up you can select 'Every' and some number of failures. The final selection is 'When the server comes back up' This will cause the action to be run at the first successful connection after being down.

1.3 Setting up a task *(actions)*

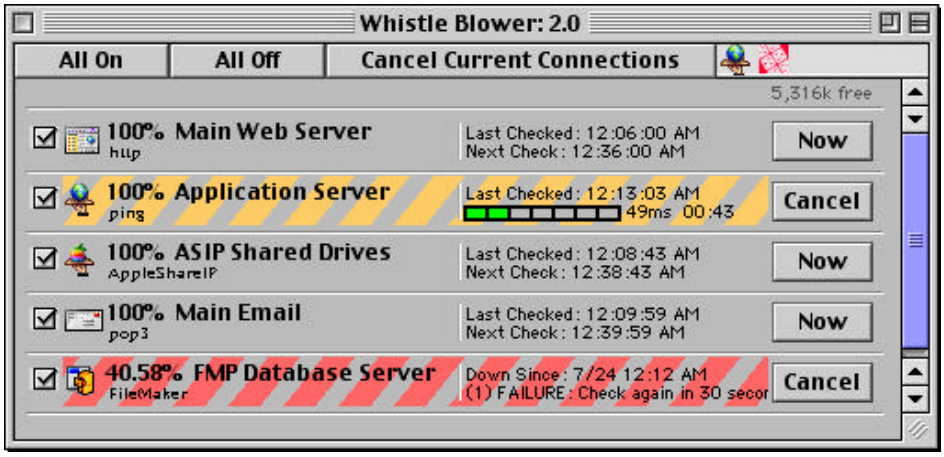
The back up selection needs some further clarification. A server is not considered to be 'down' until a regular action is performed. So if your webserver refuses one connection and then accepts the next it will not trigger the 'back up' events. If it refuses enough connections to trigger a regular action, like sending you an email, then when it comes back up the 'back up' events will be run.

After this action wait an extra: This setting at the bottom of the window is also available for all actions. It defaults to 0 but can be useful especially when restarting servers. If you know your server takes 5 minutes to reboot after a restart, this can be used to delay the next check until you know the server should be back up.

The action specific controls are displayed in the middle of the window and should be self explanatory.

You can create any number of actions for each task. You can have several task windows open at once and drag and drop actions between them, or to the desktop. If you are using the same set of actions for many servers, create them once and drag them to your desktop. This will create a text clipping containing the XML setup of the action that you can then drag and drop into any new task action windows that you need to.

2 Anatomy of the main window.



The main window holds the display for all your tasks. Each task can be enabled or disabled using the checkbox on the left hand side. Double click on a task to open it's edit window.

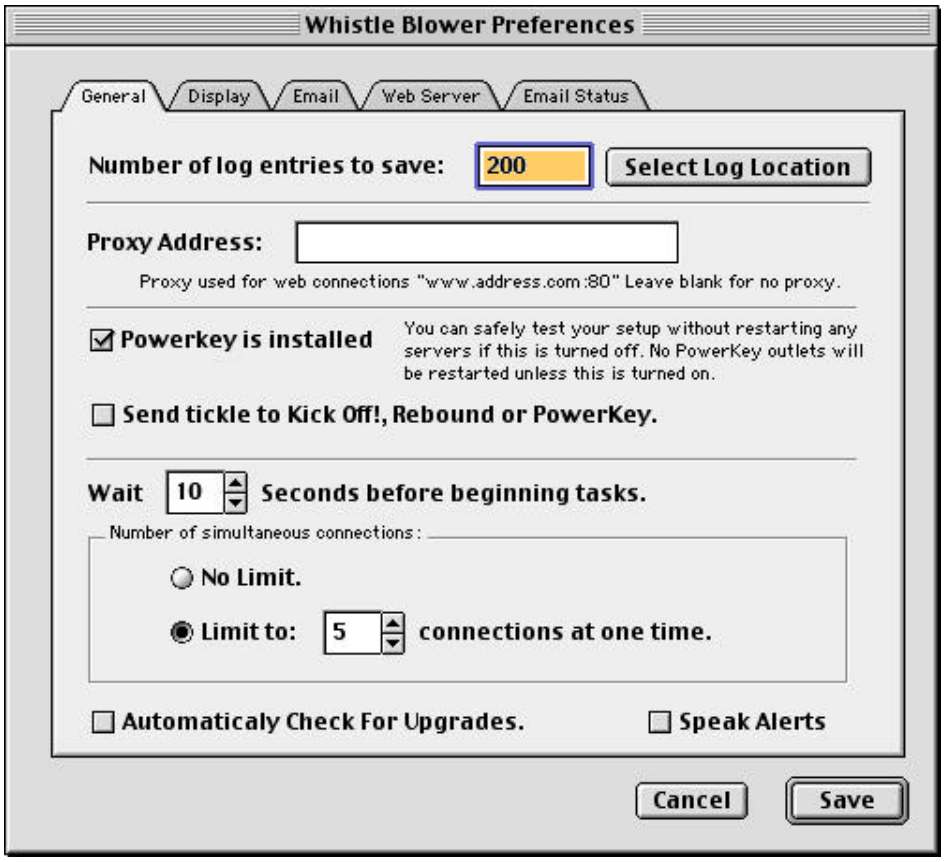
The percent uptime is now displayed for all tasks and can be reset for all the tasks by selecting 'Reset All Statistics' from the file menu, or for an individual task by highlighting it and selecting 'Reset Statistics' from the Task menu.

The left hand side of the task display shows you the name of the task and the server type. The right hand side shows you the status, the time of the last check and the time of the next check. If the task is currently being checked it's background will be highlighted with hashmarks and the connection status displayed. If the server is not responding then the hash will be red. The Last Checked message will instead display 'Down Since:' and the time when the server went down.

You can resize the columns in an individual task by clicking over the separator in the middle and dragging it to the new size. You can resize all the tasks together by doing the same, but holding down the Command key while dragging.

To re-arrange the tasks, select the one you want to move and select 'Move up' or 'Move down' from the Task Menu.

3 Setting General Prefs.



Number of log entries to save: This is the number of log entries you would like displayed in the Log window. All of these entries are cached in memory so don't use this as a monthly logfile. If you want to create a permanent or long term logfile use the 'Select Log Location' button to save a standard logfile.

Proxy Address: Whistle Blower can still connect and verify web servers even if it is running behind a proxy. Enter the proxy address here. Leave it blank if your setup doesn't require connecting through a proxy server.

Powerkey Is Installed: With this turned off you can run and test your actions to restart a powerkey and it wont actually restart it. When you are done testing you must turn this on in order for it to actually restart a powerkey.

3 Setting General Prefs.

Send tickle to KickOff!, Rebound or PowerKey: Sophisticated Circuits makes several products that can restart the Whistle Blower machine if it becomes hung. You cannot use a regular action to restart the WB machine via a powerkey, because once you turn it off, there is no process running to turn it back on again. KickOff and Rebound require no further configuration. Simply turn this on and if WB stops sending this tickle to the device, it will restart the WB box. Powerkey software 3.4 or better is required to support this feature and you must create a 'When Timer Expires' event. See your Powerkey documentation for more specific information.

Wait before beginning tasks: You may want to have Whistle Blower wait a few seconds before starting to check your servers. This will give you the opportunity to turn off servers or other actions you know are down before they have a chance to restart or send notifications.

Number of simultaneous connections: Whistle Blower can check all of your servers at the same time if you wish. But if you have many tasks this can require a considerable amount of memory. Also it can delay the response of the tasks if many are trying to check at once. This will allow you to limit the number tasks that can check at one time. The remaining tasks will wait till a spot is available and then begin their check.

Automatically check for upgrades: this setting will run the same 'Check for new versions' under the apple menu for you once a week. It sends no information at all to my servers, only requests a small text file with the most recent version number in it. If that number is greater than the version you are running it will display a dialog alerting you to the availability of the new version.

Speak Alerts: If you select this then it will automatically speak the name of the task that has gone down. You can accomplish the same thing by adding a Speak Phrase alert to all your tasks, but this can accomplish it without adding many actions.

3.1 Setting Email Prefs.

From Address: This is the return address that Whistle Blower will use when sending you emails. Use something so that you can easily recognize important alert messages.

SMTP Server Address: Whistle Blower requires access to an SMTP server in order to send email. By having 2 addresses here you will be able to send email even if the alert being sent is for the unavailability of the main one. If the first server doesn't respond within 90 seconds then Whistle Blower will attempt to send it via the backup. Again, but entering the IP address here instead of the DNS name you can still send email alerts even if your DNS server has gone down.

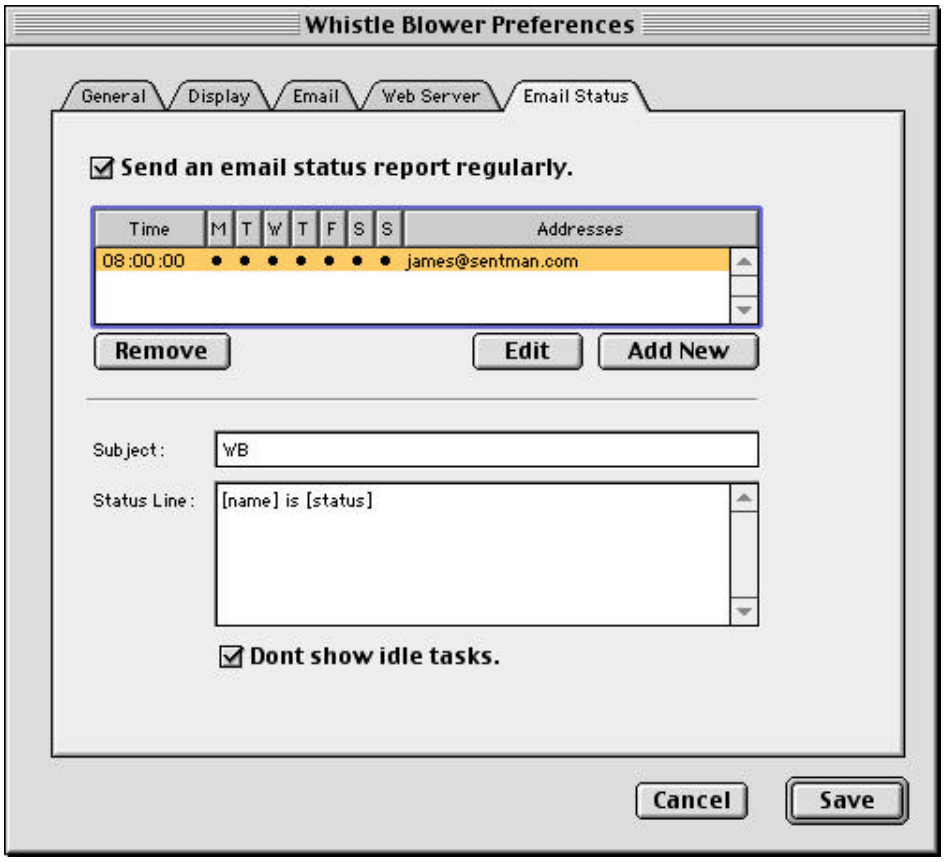
Perform Pages Via: This controls which program Whistle Blower will try to use to send pages.

3.2 Setting Web Server Prefs.

Enable Web Based Administration: Whistle Blower has extensive web based admin functions. You can pretty much control every aspect of the system from the web except for actually creating new tasks. If you have apple scripts in the scripts folder they can be launched remotely. You can also restart an attached powerkey from the web admin.

Save a Static Status Page regularly: If you wish to have a status page that can be accessible to people to whom you don't wish to give admin privs you can use this feature to save a status page into the root of a separate web server at regular intervals. it will include a meta refresh of that interval to keep it updated in the browser. The status_template.html file can be altered to fit this information in with your site design. See inside the file for details on how to set this up.

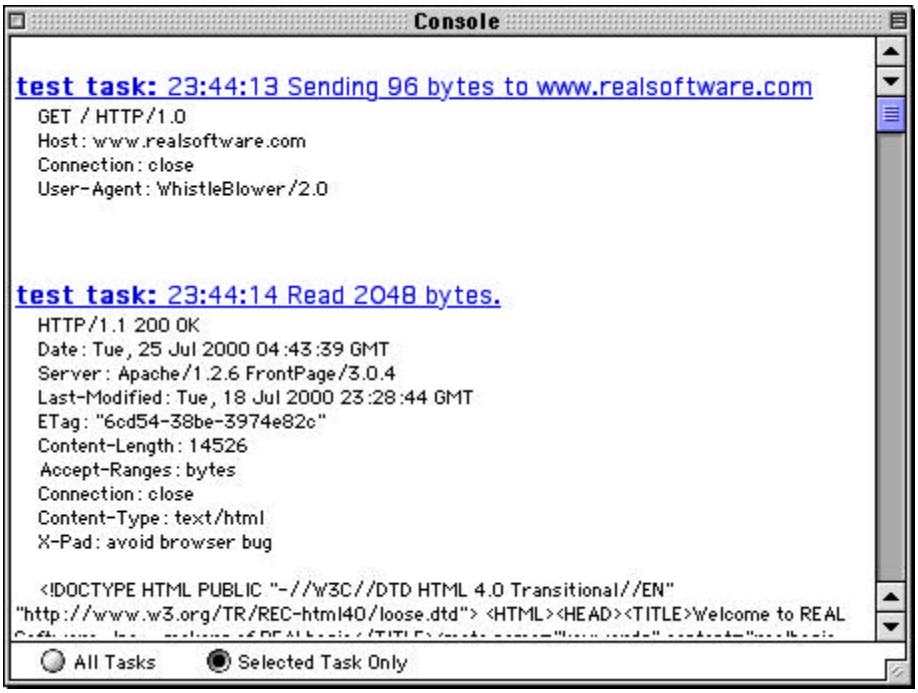
3.3 Regular Email Status Reports.



Send an Email Status Report Regularly: You can use this to send an email containing the status of all your servers on a regular basis. This is a good indication that Whistle Blower is operating properly, or just to get a weekly uptime report.

the 'Status Line' field will be repeated in the email once for each task. There are many meta tags that you can include in this field, and every other email or notification data field in the program, to fill in the proper data. There is an index to these tags in the appendix.

4.0 Using the console window.

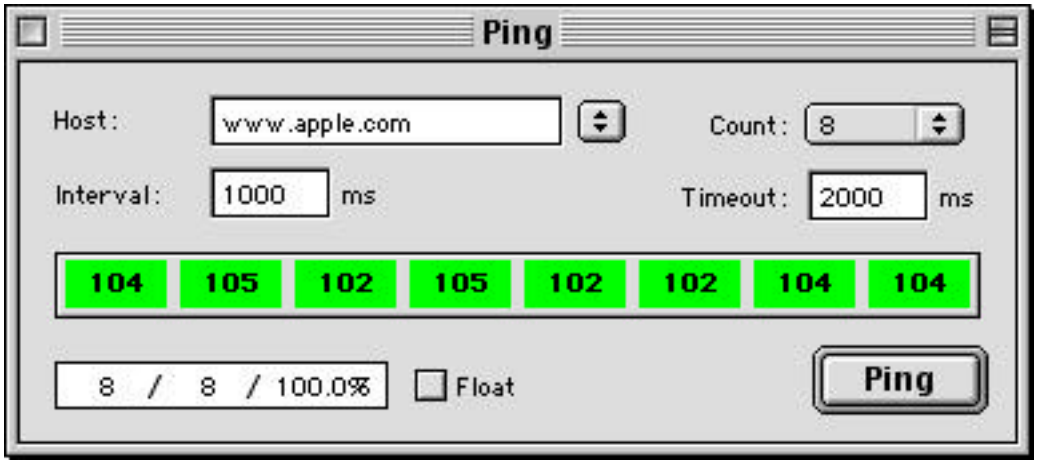


The console window can be invaluable in debugging connections. If you are trying to validate the return from a web server and not finding what you expect, this will show you exactly what is going back and forth between Whistle Blower and the server.

In order to get the proper response from a server you may need to include virtual host information or a user/password phrase. Also if you are connecting to a CGI or database you may need to include post parameters. Using the console window will show you the full conversation and any error messages or other information that the server is sending back to cause the error. This is the best way to find out why a task is not behaving as you expect.

You can open the console window at any time by selecting it from the Windows menu.

5 Sending a Ping.



You can use Whistle Blower to send a ping that is not part of a task for trouble shooting purposes. To open the Ping window select it from the Windows menu.

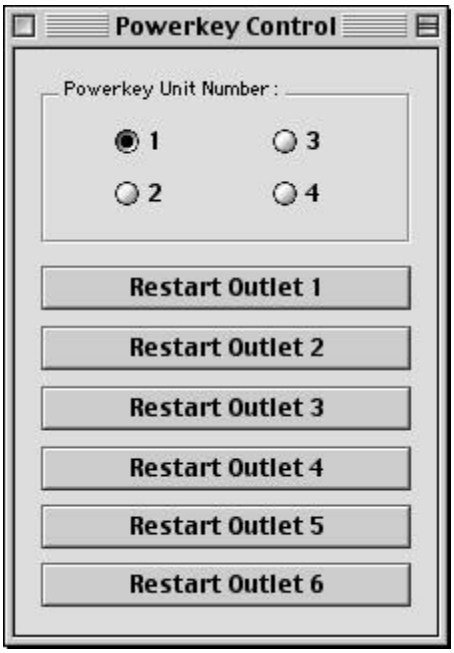
From you count menu you can select 4, 8 or continuos pinging. The display will turn yellow when a ping is sent, red if it times out and green if it is a success with the milliseconds displayed in the middle of each block.

This window was designed to be easy to see from across the room as you fiddle with patch cables under a rack of servers.

The Float checkbox will cause just the results display portion of the window to float above all other application windows on your screen. This way you can monitor a server while you work in other programs and take up a minimum of screen real estate. This is particularly useful on server machines that often have ancient small monitors attached to them.

(pinging is currently limited to PPC machines only)

6 Manual powerkey control.



You can manually restart a powerkey attached to the Whistle Blower machine with the Powerkey Control window. You'll find this under the Windows menu.

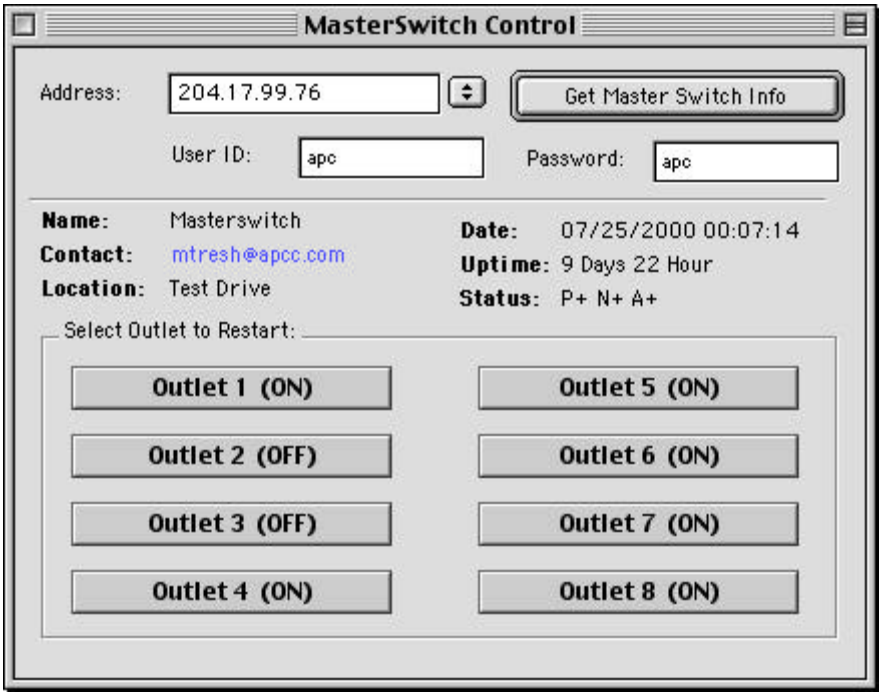
The Powerkey hardware supports up to 4 units connected to a single machine. (You may be unable to get this many units working on a single ADB buss, see the www.sophisticatedcircuits.com web site for more details) and each unit may have as many as 6 outlets.

Select the unit number and then click on the button for the outlet that you wish to restart. The outlet will be switched off for 5 seconds and then switched back on.

You cannot use this to restart the machine that Whistle Blower is running on. If you do the outlet will turn off, killing the computer and the program will not be able to turn the outlet back on again.

To restart the WB box use the Tickle feature in the prefs dialog.

7 Manual MasterSwitch Control.



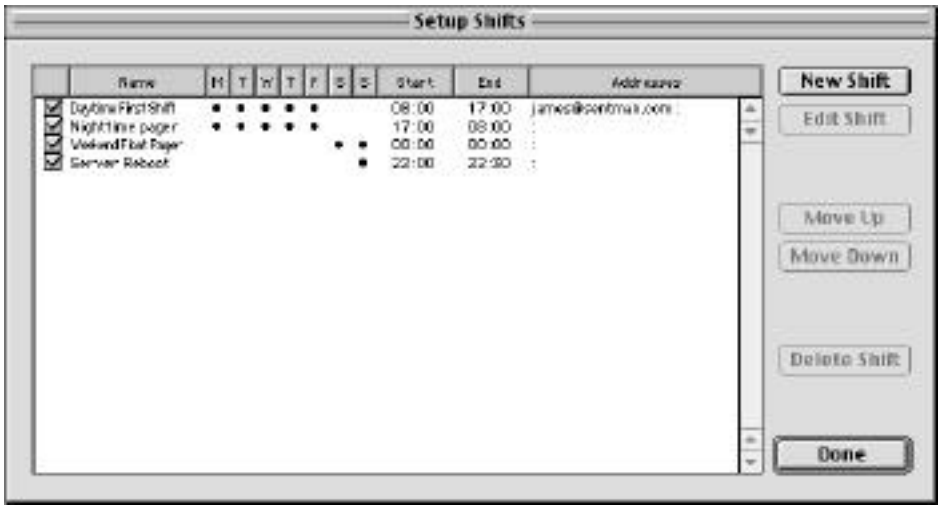
The manual MasterSwitch dialog can be opened from the Windows menu. Enter the address, UserID and Password and then click the Get Master Switch Info button to get the configuration, outlet names and status.

Each button will then display the name given to the outlet and it's on/off status.

Clicking on a button will request a restart of that outlet from the MasterSwitch using whatever timing has been set into the MasterSwitch.

Whistle Blower is compatible with both the newer 2.x versions of the MasterSwitch product as well as the earlier 1.x versions.

8 Setting up shifts.



Whistle Blower contains powerful shift management so that the alerts go to the person on duty at the time of the failure.

Shifts work in conjunction with the 'Notify Shift' action that you can add to a Task. You can select any number of shifts and only the ones that are active at the time of the alert will get the email or page.

For example, with the above shifts you would select all three of the first shifts. Whichever one is active when the action happens will get the alert, the others will not.

You can also use shifts in combination with the 'Don't Check during shift' selection for a Task. Above I have created a shift called 'Server Reboot' That is active when the server has planned downtime for a reboot. If I select this in the popup of the task then it will not try to check the server when I already know it will be down.

(Shifts are disabled in the Lite version of WB)

9 Server type details.

Where possible Whistle Blower tries to have an actual conversation with the server to verify that it is operating properly. This is not possible or not yet implemented for all server types. The server types not listed specifically here just treat a successful TCP connection to the server as an indication that the server is operating properly.

Web Servers: The web server task is the most versatile and powerful of all WB's server types. If you simply want to verify that a server is responding to requests then you can use the default setup. You should put the servers IP address in the address field and not the DNS name. This way you will still be able to check the server even if your DNS system is down. If you do this you will need to fill out the 'Virtual Host' field with the DNS name so that the server will know what home page you are requesting.

If you wish to send post parameters to verify the workings of a CGi or web database use the 'Send Post Parameters' button to configure the list. Then use the 'Return must include phrase' on the process response tab to make sure that the data is returned as you expect. You can also change the validation to 'Return must not include phrase'. I use this when verifying the availability of FileMaker databases. The Lasso code returns 'OK' for each database that should be online. If one is unavailable it returns 'Failed'. If this word is included in the output I trigger an error.

If you are trying to validate a return it may be helpful to use the console window to see exactly what is being sent back and forth.

SMTP and POP3: These server types both have a minimal conversation with the server. They wait for the 'HELO' message and then send a 'QUIT' message. This instantly frees the connection for incoming real main requests and does not cause the server to log an error as it should if the connection were simply dropped. You can also use the validation on these servers types and the console window to watch the conversation.

9 Server type details. (cont.)

Radius Authentication Servers: Radius servers are used by many ISP's to manage user accounts from a single admin system. In order to check this Whistle Blower sends a valid account request for a user called whistle blower. This will generate a 'User Unknown' error return from Radius that will let Whistle Blower know that the server is functioning properly. (This feature is PPC only at the moment)

Ping: You select the timeout and interval of pings, the number of pings that you want to send and how many must fail before considering it a failure. Pings are not important network traffic and often they will timeout even if a server is up. The default is to send 4 pings and only count that as an error if all 4 timeout. This is good if you are just checking the availability of a server. If you need to know more about server or network load you may choose to send more pings and fail if a certain percentage of them timeout. (This feature is PPC only at the moment)

Generic TCP: This task can be used to connect to any other listening port on any machine. If you need to connect to a process that is not directly supported in Whistle Blower you can try using this setting. You will need to fill in the port number that your process is using by hand. In some cases this connection may also be used for 4D servers. If the 4D task fails for your particular server try changing it to a Generic TCP task. You can also use the validation and check line by line settings for this task.

The remaining tasks are simple connect for success tasks. It will not work to try to validate the return from any other task. For future versions these tasks will be fleshed out to add more detailed conversations. More server types will also be added.

10 Meta tags in notifications.

All notification text, in emails or pages or in the status.html pages can use the same meta tags to fill in info from the task. This is useful for simplifying setting up email actions for each task. You can use a generic email message that just says '[name] is [status]' and it will be filled in with the proper information.

The tags currently supported are:

[date]
[time]
[name]
[status] 'UP' or 'DOWN'
[%uptime]
[lastchecked]
This normal displays both the label 'Last Checked:' and the date value. If the server is down then the label will change to 'Down Since:' followed by the time. This tag is special in that it includes it's own label. You don't need to have:
Last Checked: [lastchecked]
or you will end up with the label in there twice.
[Status Line]

11 Index

Whistle Blower is shareware and is offered AS IS. James Sentman makes no warranties or guarantees as to it's functionality or suitability for a particular task. While James Sentman has made every effort to deliver software of superb quality your milage may very. By using this software you agree not to hold James Sentman responsible for any data loss or server unavailability that may result. If you are unsatisfied with Whistle Blower you may return your license within 30 days for a refund of the purchase price, minus credit card and payment processing fees. Whistle Blower has a generous 45 day demo timeout. Make sure it does what you need before making your purchase.

