

## DNS, Ports, Index Files, & You

All internet traffic, whether it's a web page, an e-mail, or a chat room, is transmitted from one computer to the next in little streams of data called packets. The different types of packets – HTTP, FTP, and Telnet are just a few examples – are usually identified by a number that corresponds to a port on a computer. Savant deals with HTTP packets, which are usually associated with port # 80.

At the core, all web servers are pretty simple beasts. Savant is no exception. Savant watches packets streaming in and out of your computer, anxiously waiting for one to arrive on port # 80 that asks for a file to be sent back. That packet is known as an HTTP request because it requests a file to be sent, using the HyperText Transfer Protocol. Once Savant receives an HTTP request, it responds to it by sending the requested file or an error message.

Just like your post office address, every computer - server or not - on the Internet has a unique address called an Internet Protocol Address, or IP address for short. This address consists of four numbers, with a period between each one. For example, the IP address 161.6.18.5 belongs to a machine used for instructional purposes at Western Kentucky University. Since nobody really wants to type in a bunch of numbers to access a web site, most IP addresses have text associated with them. In the example of this server, this text is "hera.wku.edu". This text is called the domain name, and the translation between the numbers and the text is done by a special type of server called a domain name server, or DNS. Every IP address that has text associated with it is entered into a table in the DNS. This is called a DNS entry.

Your DNS entry isn't overly important to Savant, since Savant contains code to automatically detect your IP address. As long as it has your IP address, Savant can service HTTP requests to infinity and beyond. The DNS entry is much more important to you, since you want to be able to tell people an easy to remember address for your site. If you want to find out what your DNS entry is, make sure Savant is running and simply start up your web browser and in the URL line type http://localhost. "Localhost" is a special reserved keyword that tells your web browser to reference the computer it's running on. Your web browser will replace "localhost" with the DNS entry for your computer. At this point, three things can happen. If you're lucky you'll get a relatively simple DNS entry, and life will be good. If you're unlucky, you'll either get a cryptic DNS entry (like "cre734576.sdferuio.dyzes.com") or your IP address. If either of the latter two happen, you have two choices. You can live with it, or you can try one of the several free Internet services that will map your horribly ugly DNS entry or IP address to something much prettier. I highly recommend Dynamic Host Services (DHS), at http://www.dhs.org.

The only other thing to keep in mind is that some Internet service providers give you a different IP address every time you connect to the Internet. Check for this by writing down your DNS entry or IP address, restarting your computer, then comparing your new DNS entry or IP address to the old one. If they match, you're safe. If they're not, that's bad. There are two ways to deal with this problem. The first is to kindly ask your Internet Service Provider (ISP) if they'll give you the same IP address every time. If they won't, DHS has experimental software (which appears to work very well) that will give you the same DNS entry every time you connect to the Internet regardless of your IP address.

The last topic in this section is the index file. It's rare for people to type in the explicit name of a file after your DNS entry. Most just type the DNS entry, occasionally followed by a subdirectory name, and leave it at that. So what's the server do? It sends back a file called the index file. Unimaginatively enough, this file is usually named "index.html". If an index file is not found in the DNS entry or subdirectory typed in by the user, a directory listing of every file in the directory is returned to the user. For security reasons, directory listing can be disallowed so a 404 – File Not Found error is returned to the user if an index file does not exist.

# The HTTP Property Sheet



**Server DNS Entry:** This field of the HTTP property page may be left blank if you don't know what your DNS entry is, or just don't care. To determine what your DNS entry is, follow the procedure discussed in the [section on DNS](#). It's important to note that if your machine has two IP addresses you need to specify the one that is used by the Internet. This sort of thing usually happens when you have a machine acting as a proxy server between your local network and the Internet with two different IP addresses.

**Port # to Serve From:** This is the TCP/IP port that Savant will watch for HTTP requests on, and send responses to those requests on. The standard port number for the Hypertext Transfer Protocol used for web serving is 80. If you change this to a number other than 80, most web browsers will pretend that your server does not exist. If for some reason you wish to change the port number, most web browsers can be coerced into finding the server by appending a colon and the port number to the URL. Example: you change the port number to 8080. To make your web browser find it, you'll have to enter the URL `http://www.myserver.com:8080`. If you need to do this, remember that other protocols depend upon certain ports. Check to make sure the port number is free before you use it; otherwise you can prevent your computer from using other Internet functions such as e-mail and FTP.

**Index File Name:** This allows you to change the name of the default page served when someone types in your DNS entry, possibly followed by a subdirectory. By default, this is usually "index.html" or "index.htm". You can change this to whatever you want, as long as it really is the name of your index page.

**Error Message Path:** Savant allows you to create custom error message files. This field specifies the path that these error message files are located in. Custom error message files are simply an HTML file with their name being the same as the number of the error they correspond to. For example, if you create the file "404.html" in the error message path, that file will be displayed every time a "404 – File Not Found" error occurs rather than the default error message built into Savant. This allows you to customize the look and feel of your server's error messages, as well as giving you the ability to provide links to a support page or e-mail address whenever an error occurs.

# Midnight In The Gateway Of Good & Evil

Back in the dark ages of web serving, all you could place on your web page was text, with a smattering of graphics. With the introduction of faster modems and more complex web browsers in the early 1990s, a phenomenon known as "dancing baloney" began to appear. Dancing baloney were more or less completely useless technologies, such as animated GIFs, designed to attract people to web pages. In their quest for new and improved dancing baloney, corporations began to look high and low for technologies that could be subverted.

For several years, the National Center for Supercomputing Applications (NCSA) at the University of Illinois had been refining an idea called a gateway interface. A gateway interface is simply a way to run a piece of software on a server using data from a client, and returning the data to the client. The NCSA's work centered around developing a special type of gateway interface that allowed a web browser to run software on a web server. This specific type of gateway interface became so popular that it was called the Common Gateway Interface, or CGI.

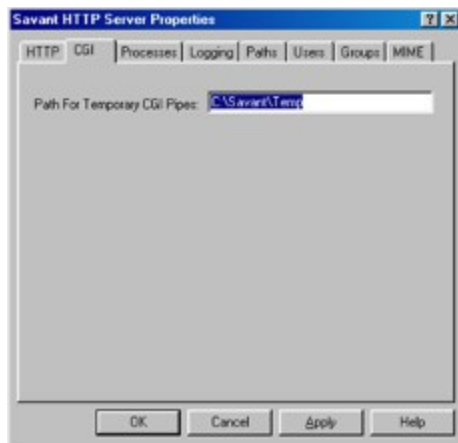
Originally designed to be useful for highly technical applications, CGI was subverted into the ultimate dancing baloney. Hit counters, HTML postcards, chat rooms.... the list goes on. Although hardcore Internet types generally frown upon "dancing baloney", it can be a lot of fun to play around with on your own web server. If you're into programming, it's fairly easy to write your own custom CGI programs and scripts. For more information and examples, consult the NCSA web site at <http://hoohoo.ncsa.uiuc.edu>. If you don't want to bother with writing your own CGI programs and scripts, the web is chock full of web sites giving them away.

The Common Gateway Interface is fast approaching a major turning point. Several vendors, notably Microsoft, have developed extensions to the old standard and defined new standards such as Active Server Pages (ASP) and ISAPI scripting. These new interfaces have two major advantages over the CGI standard on Windows platforms – they're slightly faster, and you can do a lot more with them. They do have some severe drawbacks – they're extremely complicated, require expensive software packages like Microsoft Visual Studio to create, have notorious security flaws, and for the most part only run on Windows. Something of a war is now being waged, with supporters of standard CGI (mostly users of large-scale systems, freeware & shareware advocates, and programmers) on one side and supporters of ASP and ISAPI (mostly Microsoft & friends) on the other. Which standard will eventually win out is still undecided. In the mean time, Savant explicitly supports the CGI 1.1 standard, WinCGI 1.3a, and experimentally supports ISAPI applications (not filters!). Microsoft used to supply add-on extensions for ASP and FrontPage that would work with Savant - they have stopped making these packages available in the past year. If you want to use ASP at this point in time, you're pretty much locked into either using one of Microsoft's servers or spending a large amount of money on proprietary add-ons such as the ones sold by ChileSoft.

Savant uses the standard NCSA implementation of CGI, which works with any properly written CGI program or CGI script. A CGI program is an executable file, while a CGI script is a text file that is interpreted by an executable file. By far the most common scripting language for CGI scripts is Perl. Savant automatically installs the latest distribution of Perl5 so all you have to do is write the script. For security reasons, CGI programs and scripts must be placed in a directory allowing CGI execution. For more information on how to create a CGI directory, see the Savant Help section on Paths. A step-by-step guide to setting up CGI scripts and programs is in the How Do I... section.

On to the evil. CGI is a notorious security problem on UNIX-based web servers because it's easy for hackers to exploit a poorly written CGI program or script. Fortunately, the common hacks used on UNIX web servers have no effect on Windows-based web servers. Can the CGI in Savant be hacked? Possibly, since there's no such thing as a "non-hackable" server. Is it likely? Not really. Although you should be aware that it is possible, I wouldn't lose sleep over it.

# The CGI Property Sheet



**Path for Temporary CGI Pipes:** CGI uses a special type of file known as a *pipe* to send data from a web browser to the server and back again. This field lets you decide where these special files are kept. If you want to have blazing fast CGI performance and you have RAM to spare, I suggest you place the path of a RAM drive here.

## What Has Your Process Done For You Lately?

A long, long time ago, PCs could only run one program at a time. Then came the early versions of Microsoft Windows, which allowed you to run more than one program at a time (called multitasking) by doing some clever cheating. Instead of letting one program have full control until it was finished, it gave each program a chunk of time in which it got control. Technically it still only ran one program at a time, but as far as the user was concerned, it looked like the programs were running at the same time.

Windows 95, 98, NT 4, and higher provide "true" multitasking – multiple programs run at the same time without having to cheat. Because of this, programs can split themselves into small tasks, called processes, and have all of these processes run at the same time. Savant does this as a way to serve HTTP requests as efficiently as possible. Time for an example: suppose 25 people want to look at your web site, and all 25 HTTP requests are sent at about the same time. The simple way to handle this situation is to have Savant service the requests one at a time, moving from the first to the last in sequential order. This is fine if you're the first one served, but there's going to be a noticeable pause if you're 25th in line. So instead, Savant creates 25 processes that each act like an individual copy of Savant, and serves all 25 requests at the same time. While complicated, this provides extremely fast and reliable web serving. The [Processes Property Sheet](#) is the interface through which you control how Savant creates and handles processes.

# The Processes Property Sheet



**Initial Number of Processes:** Most web browsers have the capability of receiving more than one file at a time. This greatly speeds up the download time of an entire web page (including graphics). As a rule, one process serves one file in Savant. This field contains the number of processes that Savant initially starts. If Savant needs more processes, then it will create more. The default of 5 should be sufficient for most web servers. If your server experiences periods of inactivity followed by high-traffic bursts, you may want to increase this number to make response time faster during the high-traffic bursts.

**Maximum Number of Processes:** This is the maximum number of processes Savant will create. Effectively, this is the number of clients your web server can handle simultaneously. If more people request connections than this number allows, they will be refused connections. By setting this field to a reasonable number you can prevent malicious people from locking up your computer by creating thousands and thousands of server processes. This field's initial value is high enough for a moderate traffic web server, but should probably be increased to a number between 100 and 200 for higher traffic sites.

**Number of Processes to Keep Free:** Savant will create more processes when there are less free processes than the number in this field. The default number for this field should be OK for most situations – you may need to increase it if your server has high bursts of traffic as described above in "Initial Number of Processes". Be careful to keep this number much smaller than the Maximum Number of Processes, otherwise you will prevent users from being able to connect to your server.

**Process Compact Period:** After Savant responds to an HTTP request by sending a file, it will keep the process it created alive for the number of seconds specified in this field. This is based on the theory that most web pages are going to be read in only a few minutes, and then a link leading to another page on the same server will be clicked by the user. By keeping the process alive, Savant can save time and processing by responding to HTTP requests from the same user by using the same process. This keeps Savant from constantly terminating and creating processes to serve requests from a single user. Some web browsers are extremely well mannered, and send Savant a message when the user will no longer be using the process. When Savant receives this message, it will immediately terminate the process. Otherwise, it will wait for the number of seconds specified in this field. The default value of 3000 seconds (5 minutes) is usually more than sufficient. If you have a high-traffic web site, you might want to reduce this value to as little as 30 seconds.

**Process Compact Laziness:** As with every other aspect of computing, Murphy's Law holds true for web serving. Sometimes a process can go horribly awry – usually the result of a web browser crashing while somebody's looking at your web site. This leaves a process trying to send a file to a client that doesn't

exist. The process won't terminate after the number of seconds specified in the Process Compact Period field, because it's still doing something. Of course, what it's doing will never be finished, since the client it was sending a file to no longer exists.

To keep these useless processes from accumulating and eating up your computer's memory and processing resources, Savant has a special watchdog process that checks every process to make sure it's doing something useful. If a process isn't doing something useful, the watchdog process terminates it. This field lets you set how often the watchdog process checks every process to make sure they're all being well behaved. Time to whip out the calculators:  $\text{Process Compact Period} \times \text{Process Compact Laziness} = \# \text{ of seconds the watchdog process checks every process}$ . The default values for these two fields send the watchdog process out every 10 minutes. If most of the traffic to your web site is via dial-up services that have a tendency to disconnect users, you might want to decrease this value so these useless processes are terminated more often.

## It's Big, It's Heavy, It's Wood!

Ok, Ok. So it's not heavy, and it's not wood. (Apologies to *Renn & Stimpy*.) But it is "Big", and it is "Better Than Bad". Logging is the practice of keeping a special record of computer events. In the case of Savant, the "events" are people looking at stuff on your web site. The log files are simple text files – you can look at them in Windows Notepad, your favorite word processor, or even the MS-DOS command line. Taken as a whole, the log files created by Savant allow you to see who looked at what on your web site when. This can help you to identify popular areas of your web site, and areas that nobody bothers with. It can also give you an idea of who's looking at your site, and how you can target your content to attract them. Savant creates three log files – the General Log keeps a transaction-by-transaction record of every file request made to Savant; the Hit Log keeps a record of how many times each individual file on your site was accessed; and the Reference Log keeps track of which clients have accessed files, and how often they've accessed them.



# The Logging Property Sheet



**Path To Store Logs At:** This is the directory on your computer where Savant will store the log files. By default, this will be a subdirectory of the directory you installed Savant in named "Logs". (Usually C:\Savant\Logs)

**General Log:** The General log has the highly imaginative default name of "general.txt". The General log is just that – it keeps a sequential record of every HTTP request that Savant receives. It's composed of entries that follow the format of:

```
somebody.rhh.wku.edu - - [03/Aug/1998:21:19:32 +0600] "GET /index.html" 200 1337
DNS Entry - - [Day/Month/Year:Time (GMT) +0600] "Method /filename" error code bytes
```

The *DNS Entry* field lists the DNS entry (or IP address if they don't have a DNS entry or you disable *Reverse DNS Lookup*) of the person looking at your web site. The date and time fields are pretty self-explanatory; the time is in your local time zone. The +0600 (or something like that, depending on your time zone) is a reminder that Greenwich is 6 hours ahead of the Central Time zone in the US. (I live in the Central Time zone.) The *Method* field lists the type of method the web server used to request the data, while the *filename* field gives the location and name of the file that was requested. *Error code* is the error code returned by Savant upon completion of the request – an error code of 200 means that the request was served properly with no problems. The *bytes* field gives the number of bytes that were sent by Savant to complete the transaction.

**The Hit Log:** The Hit log keeps a record of how many times each file on your web site has been accessed, and the time of the last access. Each file on your web site that has been accessed by a client has an entry in the Hit log of the following format:

```
/index.html=2 03/Aug/1998:21:19:38 +0600
filename = number of accesses date:time +0600
```

*Filename* is the location and name of the file that the log entry is for, *Number of Accesses* is the number of times the file has been successfully served to a client, and the date and time are self-explanatory. Optionally the Hit log can keep a count of the number of connections made to your web site, the number of files successfully sent, and the number of kilobytes served.

**The Reference Log:** The Reference log keeps a record of every file accessed on your web site, arranged by file name. Each file on your web site has an entry, and below it is a listing of every DNS entry or IP address that has accessed the file, along with the total number of accesses made from that location and the time of the last access. An example for the file "index.html" is:

[/index.html]  
somebody.rhh.wku.edu=2 03/Aug/1998:21:19:38 +0600  
[/filename]  
*DNS Entry = Number of Accesses Date:Time of Last Access +0600*

This is a nifty log, because you can see what one particular person is looking at on your web site, and how often they're looking at it.

## Here An HTTP Path, There A FAT Path

Anybody who's ever used a web browser knows that the way files are arranged on a web site is different from the way files are arranged on a PC running Microsoft Windows. Instead of having a DOS-style root directory (such as "C:\"), web sites have a root consisting of the web server's DNS entry followed by a slash (not a backslash!) In fact, web servers have no backslashes anywhere in their paths. Paths are simply the locations of files on computers; "C:\Windows\notepad.exe" is an example for a computer running Windows. All of this leads to a pretty obvious problem – how do you make Windows-style paths (properly known as FAT Paths) look like web server paths (properly known as HTTP Paths)?

Lucky for you, Savant does all of the work for you. All you have to do is give it a FAT path on your computer, and what you want the HTTP path to be called. Whenever someone connects to your web server and types in a file path, Savant translates back and forth between FAT and HTTP paths. Only directories that you enter in the Paths section of the Savant configuration utility will be served as a security measure (you don't really want to serve your entire hard drive, do you?) This also lets you do some nifty tricks, like serving the CD in your CD-ROM drive. If you're connected to a Local Area Network (LAN) with Microsoft Networking, you can even serve shared directories on other computers.

# The Paths Property Sheet



**Creating A New Path:** To create a new path, click on the *New Path* button in the **Paths** property page of the configuration utility. A dialog box will appear, with the following fields:

Local Path: This is the directory (FAT Path) on your computer that you wish to serve. Make sure you type in a complete path – "C:\WebServer" is acceptable, "WebServer" and "\WebServer" are *not* acceptable. If you want to browse for the path you wish to serve, click on the small button with three periods in it.

HTTP Path: This is the directory that you wish *Local Path* to be served as. It doesn't have to have the same name as *Local Path* – any name you wish is acceptable. The only restrictions are that the path must start with a slash (***NOT*** a backslash!!) For example, "/coolstuff" is legal, while "\coolstuff" and "coolstuff" are not legal. As a general rule, most HTTP paths are composed entirely of lowercase letters. This is a rule you should follow – it makes life easier for everyone.

Scripting: These radio buttons let you allow or disallow certain kinds of script and program execution from a particular directory. If you want to run CGI scripts or programs from this directory, be sure to check the *CGI* radio button. The same is true for WinCGI scripts and ISAPI applications. Check the *Normal* radio button to disallow CGI execution from the directory. As Savant is currently written, you can't have different kinds of scripts in the same directory (in other words, you can't put ISAPI applications in the same directory as your CGI scripts.) You should always place your CGI programs and scripts in a separate directory from your HTML. Placing HTML in a CGI directory is a very bad idea – the HTML will not be served as a security precaution.

Allow Directory Listing: When you enable this option, Savant will return an FTP-style file listing to a web browser if an index file does not exist in the directory. This option is simply a security feature – if you don't want people to see a list of files in a directory, disable it. This option is ignored if the directory serves scripts or applications.

Group or User: This drop-down box lists every user and group on your web server. To restrict access to a directory to a particular person or group of people, click on their user name or group name in this drop-down. Before anyone can view the contents of this directory, they must input the valid user name and password you have chosen. If you choose "Anyone" from the list, anybody can see the contents of the directory. This is the recommended default, since usually you don't want to restrict access to areas of your web site.

Accessible Location: This option restricts directory access to a certain group of users based on their IP address. As you recall from previous sections of this User's Guide, every computer on the Internet has an IP address that consists of four numbers, separated by periods (127.0.0.1 is an example). Everybody on

a Class A Subnet shares the same first number of the IP address. Everyone on a Class B Subnet shares both the first and second numbers of the IP address, and so on for Class C and D subnets. To make the directory available to the world at large, set this field to the "Anywhere" option. If you're a member of a large corporation or ISP, the Class A Subnet option will probably limit access to people who are in the same corporation or use the same ISP as you. For all but the largest networks and ISPs, the Class B Subnet option will probably do the same thing. For example, every IP address on the campus of Western Kentucky University begins with the same first two numbers. To limit access of a directory to just WKU's campus, I could set this field to the "Class B Subnet" option. Because everyone in my dorm shares the same third number in the IP address, I could restrict access to just my dorm by selecting the "Class C Subnet" option. A Class D Subnet shares all four of the numbers, effectively limiting it to the server itself. This is a nifty tool if you want to test something on your web server, but don't want to make it available to anyone else until you're sure it works.



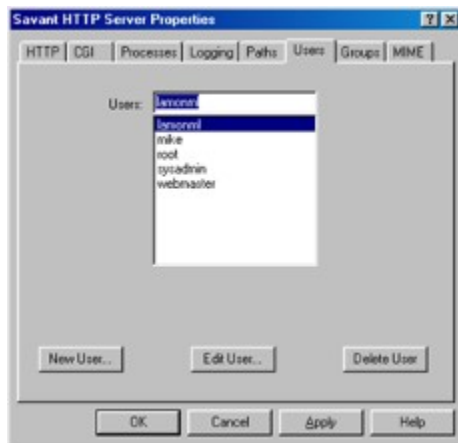
**Editing a Path:** To edit the information for a path that already exists, simply click on the path's name in the *Paths* list box, then click on the *Edit Path* button. All of the information for the directory will be placed in the appropriate fields. For a full explanation of each field, see the above section on "Creating a New Path".

**Deleting A Path:** To delete a path, click on the path's name in the *Paths* list box, then click on the *Delete Path* button. If you accidentally hit the delete button, you can "undo" the deletion by clicking on the *Cancel* button, which will discard all changes you've made in the configuration utility since either the last time you clicked *Apply* or when you launched the configuration utility.

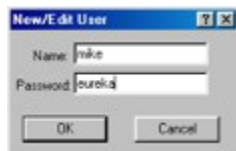
## Restricting Server Resources

On occasion, you might have an area of your web site that you don't want to be visible to the world at large. Savant allows you to create users (just like an e-mail or FTP server) with their own passwords that you can restrict access to. If you associate a user with a certain directory (as described in the Paths section of this User's Guide), a dialog box will pop up asking for a user name and password anytime someone tries to look at that directory. For information on how to create a group that allows multiple users access to a directory, see the section on Groups.

# The Users Property Sheet



**Creating A New User:** To create a new user, click on the *New User* button in the property page labeled "Users". A dialog box will appear, asking you for a user name and a password. Except for length, there are no restrictions on the user name or password. Neither is case-sensitive, so the usage or non-usage of capital letters doesn't matter.



**Editing A User:** To edit a user's name and password, click on the user's name in the list box, then click on the *Edit User* button. The dialog box that appears contains the values for the current user's name and password. After you edit these fields to your satisfaction, press the *OK* button inside the dialog box to save your changes.

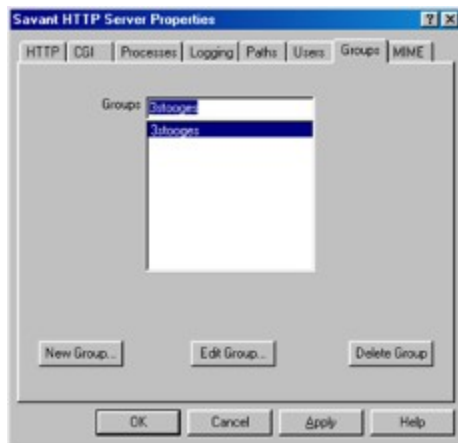
**Deleting A User:** To delete a user, click on their name in the list box, then click on the *Delete User* button. If you accidentally delete a user, you can click on the *Cancel* button to discard the changes you've made inside the configuration program.

## Classifying Users

On occasion, users on your web server will fall into certain categories. For example, users may be a certain age, come from a certain city, or live in a certain building. Whatever they have in common, you can place these users together in a group. You can give this group (see the Paths section) access to a directory. For example, you can grant registered users of your software access to a directory containing software upgrades.



# The Groups Property Sheet



**Creating A New Group:** To create a new group, click on the *New Group* button in the *Groups* property page. A dialog box will appear, with every user on your server in a list box on the left-hand side and an empty list box on the right hand side. Between the two boxes appear two buttons, with arrows pointing left and right. To make a user a member of the group you are creating, highlight their user name by clicking on it with the mouse, then click the button pointing to the right hand list box. Their user name will disappear from the list of users, and reappear in the list of group members. If you want to remove a user from the list of group members, highlight their name, then click the arrow pointing to the left-hand list box. Don't forget to pick a name for the group, and type it in the field labeled *Group Name*.



**Editing A Group:** To edit a group, highlight its name in the list box by clicking on it. Then click on the *Edit Group* button to display a dialog box like the one above. This dialog box contains all of the data about the group. You may add and remove users from the group or change the group's name in this dialog box.

**Deleting A Group:** To delete a group in Savant, click on the group's name in the list box to highlight it. Then click on the *Delete Group* button. If you delete a group by mistake, you can click on the *Cancel* button to discard all of the changes you've made since you started the configuration utility or last clicked the *Apply* button.

## MIME Types & You

In the early days of the Internet, the World Wide Web didn't exist. The most common medium of information exchange was e-mail. If you wanted to send a picture or a sound file to someone, you had to e-mail it to them as an attachment to a regular e-mail. This got really confusing really fast – e-mail programs of the time treated every non-text file as a simple binary file. When you received a file, neither you nor your e-mail program had any idea if it was a picture, a sound, or an executable program.

To solve this problem, the Multimedia Internet Mail Extension, or MIME, standard was devised. Under the MIME system, a small string of text is sent along with the file. This string of text consists of two parts separated by a slash. The first part gives a general grouping for the file, for example "text", "image", or "audio". The second part gives a specific type of file, for example "html", "gif", or "midi". These identification strings are standardized by the World Wide Web Consortium, so there isn't any confusion as to which string goes with which file.

For obvious reasons, the MIME standard is very important to the World Wide Web. A web browser has to know what type of file it's looking at so it can see what to do with it. Graphic images and text are usually displayed by the browser, while more complex file types like ZIP, PDF, and MIDI are viewed outside of the actual web browser by programs that the browser launches.

The web browser keeps track of what type of file it is based on its MIME type, sent by the web server. If you want to place a certain type of file on your web server, be sure to check that its MIME type is supported by Savant. If it isn't, use the graphical configuration utility to add it.

# The MIME Property Sheet



**Creating A New MIME Type:** To add a MIME type to Savant, click on the *New MIME* button in the *MIME* property page of the graphical configuration utility. In the dialog box that appears, place the extension of the file type (the letters that come after the period in the file name. For example, the extension of GUIDE.PDF is PDF) and the MIME description.



**Editing A MIME Type:** To edit a MIME type, highlight it by clicking on it with the mouse in the list box. Then, click on the *Edit MIME* button. A dialog box will appear, containing the extension and MIME description. Edit these fields to your satisfaction, then click the *OK* button.

**Deleting A MIME Type:** To delete a MIME type, highlight it by clicking on it with the mouse in the list box. Then, click on the *Delete MIME* button. If you delete a MIME type by accident, you can click on the *Cancel* button to discard all of the changes you've made since you started the configuration utility or the last time you clicked the *Apply* button.

# Common MIME Types & Descriptions

Below is a table of common MIME types with their associated descriptions and file extensions. If you'd like your server to support a file type that Savant does not natively support, find its information in the below table and use the [MIME Property Sheet](#) of the configuration utility to add it to Savant.

File types that Savant provides native support for are in *italics*.

<b><u>MIME Type</u></b>	<b>MIME Description</b>	<b>MIME Extension</b>
<i>Sun Microsystems Audio</i>	<i>audio/basic</i>	<i>AU</i>
<i>Binary Application</i>	<i>application/octet-stream</i>	<i>BIN</i>
<i>Windows Bitmap</i>	<i>image/bmp</i>	<i>BMP</i>
<i>Microsoft Word Document</i>	<i>application/msword</i>	<i>DOC</i>
<i>Encapsulated PostScript Application</i>	<i>application/postscript</i>	<i>EPS</i>
<i>CompuServe GIF Image</i>	<i>image/gif</i>	<i>EXE</i>
<i>Windows Help File</i>	<i>application/winhelp</i>	<i>GIF</i>
<i>Hypertext Document</i>	<i>text/html</i>	<i>HLP</i>
<i>Hypertext Document</i>	<i>text/html</i>	<i>HTM</i>
<i>JPEG Image</i>	<i>image/jpeg</i>	<i>HTML</i>
<i>JPEG Image</i>	<i>image/jpeg</i>	<i>JPG</i>
<i>JavaScript</i>	<i>application/x-javascript</i>	<i>JPEG</i>
<i>Microsoft Access Database</i>	<i>application/x-msaccess</i>	<i>JS</i>
<i>MIDI Music File</i>	<i>audio/midi</i>	<i>MDB</i>
<i>MIDI Music File</i>	<i>audio/midi</i>	<i>MID</i>
<i>QuickTime Video</i>	<i>video/quicktime</i>	<i>MIDI</i>
<i>MPEG Layer 3 Audio</i>	<i>audio/x-mpeg</i>	<i>MOV</i>
<i>MPEG Movie</i>	<i>video/mpeg</i>	<i>MP3</i>
<i>MPEG Movie</i>	<i>video/mpeg</i>	<i>MPE</i>
<i>MPEG Movie</i>	<i>video/mpeg</i>	<i>MPEG</i>
<i>Adobe PDF</i>	<i>application/pdf</i>	<i>MPE</i>
<i>PowerPoint Presentation</i>	<i>application/vnd.ms-powerpoint</i>	<i>PDF</i>
<i>PowerPoint Presentation</i>	<i>application/vnd.ms-powerpoint</i>	<i>POT</i>
<i>PowerPoint Presentation</i>	<i>application/vnd.ms-powerpoint</i>	<i>PPS</i>
<i>Adobe PostScript</i>	<i>application/postscript</i>	<i>PPT</i>
<i>Microsoft Publisher</i>	<i>application/x-mspublisher</i>	<i>PS</i>
<i>QuickTime Movie</i>	<i>video/quicktime</i>	<i>PUB</i>
<i>RealAudio/RealVideo</i>	<i>audio/x-pn-realaudio</i>	<i>QT</i>
<i>RealAudio/RealVideo</i>	<i>audio/x-pn-realaudio</i>	<i>RA</i>
<i>RealAudio/RealVideo</i>	<i>audio/x-pn-realaudio</i>	<i>RAM</i>
<i>Rich Text Format Document</i>	<i>application/rtf</i>	<i>RPM</i>
<i>Basic audio</i>	<i>audio/basic</i>	<i>RTF</i>
<i>TAR Archive</i>	<i>application/x-tar</i>	<i>SND</i>
<i>LaTeX Document</i>	<i>application/x-tex</i>	<i>TAR</i>
<i>TIFF Image</i>	<i>image/tiff</i>	<i>TEX</i>
<i>TIFF Image</i>	<i>image/tiff</i>	<i>TIF</i>
<i>Plain text</i>	<i>text/plain</i>	<i>TIFF</i>
<i>Waveform Audio</i>	<i>audio/x-wave</i>	<i>TXT</i>
<i>Wavelet Image</i>	<i>image/wavelet</i>	<i>WAV</i>
<i>Windows MetaFile</i>	<i>application/x-msmetafile</i>	<i>WI</i>
<i>Windows Write Document</i>	<i>application/x-mswrite</i>	<i>WMF</i>
<i>Excel Spreadsheet</i>	<i>application/vnd.ms-excel</i>	<i>WRI</i>
<i>ZIP Archive</i>	<i>application/zip</i>	<i>XLS</i>
		<i>ZIP</i>

# Couldn't Bind Socket

## CRITICAL!

Savant returns this error when it cannot associate its server socket with the port specified in the [HTTP Property Sheet](#). By default, this port is 80. Since Savant cannot serve HTTP requests without a server socket to receive the requests, Savant shuts down when this error occurs.

Most often, this error is caused by trying to simultaneously run two web servers. If such is the case, the web server that was initially running will be unaffected. It's also possible that an application that was using port 80 didn't properly release the port, or your TCP/IP service did not reset the port properly.

To solve this problem, try the following steps:

1. Verify that only one copy of Savant is being run. To make sure Windows didn't remove the icon from the system tray without terminating the program, press CTRL-ALT-DEL and make sure that Savant does not appear on the Task List. If it does, highlight it with a left click of the mouse, then click on the End Task button. Wait until verification appears that Savant has been terminated, then try running it again.
2. Verify that Savant is the only web server you are currently running. This error will occur if you're running any other web server (including Microsoft's Personal Web Server and Internet Information Server, Apache, NCSA's HTTPd, and Inprise's application server.) If you find another server, shut it down and try restarting Savant. Please note that some applications (such as ICQ, Gnotella, and others) are sneaky and run their own little generic server on port 80 if you don't explicitly turn it off.
3. Try restarting your computer to be sure another application has not failed to release port 80 for use by Savant.
4. Re-install your TCP/IP implementation in the Network dialog of the Control Panel to correct a possibly corrupted TCP/IP implementation.

# Failure To Create Asynchronous Event

## CRITICAL!

Savant was unable to set up its server socket to handle simultaneous requests. Simultaneous requests to Savant would cause the socket to block, possibly crashing the client browsers as well as the computer running Savant. For this reason, Savant shuts down when it encounters this error.

This error is usually caused by a TCP/IP fault or a lack of available memory.

To solve this problem, try the following steps:

1. Restart your computer to correct a TCP/IP fault.
2. Verify that you have enough available memory for Savant to operate. Savant can run on a system with as little as 4 megabytes of RAM, but at least 12 megabytes is recommended.
3. Re-install your TCP/IP implementation in the Network dialog of the Control Panel to correct a possibly corrupted TCP/IP implementation.

# Failure To Create Socket

## CRITICAL!

Savant was unable to create a server socket. Since Savant is unable to serve requests without a server socket, Savant shuts down.

This error is usually caused by either a lack of available memory to create a socket, or a TCP/IP fault.

To solve this problem, try the following steps:

1. Restart your computer to correct a TCP/IP fault.
2. Verify that you have enough available memory for Savant to operate. Savant can run on a system with as little as 4 megabytes of RAM, but at least 12 megabytes is recommended.
3. Re-install your TCP/IP implementation in the Network dialog of the Control Panel to correct a possibly corrupted TCP/IP implementation.

# Failure To Start Listening Socket

## CRITICAL!

Savant was unable to begin listening for requests on its server socket. Since Savant cannot respond to requests when this error occurs, Savant shuts down.

This error is caused by a TCP/IP fault.

To solve the problem, try these steps:

1. Restart your computer to correct a TCP/IP fault.
2. Re-install your TCP/IP implementation in the Network dialog of the Control Panel to correct a possibly corrupted TCP/IP implementation.



# Failure To Start WinSock

## CRITICAL!

Savant was unable to initialize the Microsoft Windows TCP/IP service. Web serving requires the use of a TCP/IP service, so Savant shuts down.

This error is caused by the absence of a valid TCP/IP service or a lack of memory.

To solve the problem, try these steps:

1. Restart your computer to correct a possible TCP/IP fault.
2. Verify that you have enough available memory for Savant to operate. Savant can run on a system with as little as 4 megabytes of RAM, but at least 12 megabytes is recommended.
3. Verify that you have a recent version of WinSock installed on your computer. The latest release is available free from Microsoft at [www.microsoft.com](http://www.microsoft.com).
4. Re-install your TCP/IP implementation in the Network dialog of the Control Panel to correct a possibly corrupted TCP/IP implementation.

# Failure Waiting For An Available Thread

## CRITICAL!

Savant received an error message from the operating system when it tried to start a new process. Since Savant cannot serve requests without the ability to create new processes, Savant shuts down upon receipt of this error.

This error is caused by an operating system fault. As a general rule, this error usually occurs after an operating system crash.

To solve this problem, restart your computer.

## Network Down

### CRITICAL!

Savant was unable to detect a valid network connection. Savant requires a valid TCP/IP network to start up. Once Savant has started, it will continue to run regardless of network service fluctuations. In other words, Savant will not start without a valid TCP/IP network but if your network goes down then comes back up Savant will continue to service requests normally. Since Savant cannot properly initialize its services without a valid TCP/IP network connection, Savant shuts down upon receipt of this error.

This error is almost always the fault of your network connection. It can sometimes be caused by a TCP/IP fault.

To solve this problem, try the following steps:

1. Restart your computer to clear any existing TCP/IP fault.
2. Verify that your TCP/IP network connection is valid.
3. Re-install your TCP/IP implementation in the Network dialog of the Control Panel to correct a possibly corrupted TCP/IP implementation.

## Process Snafu

### CRITICAL!

The Windows kernel has faulted and is sending Savant faulty information about the status of processes. Since Savant has lost control over its processes, it shuts down upon receipt of this error.

This error is caused by a 32-bit Windows operating system kernel, not Savant.

To solve this problem, restart your computer.

# Registry Has Been Corrupted.

## CRITICAL!

Savant has detected an error in its registry entries. Since Savant cannot load its start up information, it shuts down.

This error is usually caused by the volatility of the Windows registry, which on rare occasions loses information. Savant is highly dependent upon accurate registry information.

In older versions of Savant, it was recommended that the server be uninstalled and then reinstalled, since the installation process wiped the old registry entries entirely. The newer versions of Savant are designed to import registry data, so the corrupted registry entries will not be wiped. Instead, perform the following procedure:

1. Uninstall Savant.
2. Run the regedit command from the Run option of the Windows Start menu.
3. Locate the registry path `\\HKEY_LOCAL_MACHINE\\Software\\DAEMONS\\Savant` and delete it from the registry.
4. Re-install Savant.

Let me stress that this should only happen on extremely rare occasions due to some improved code – it is much less likely to happen than with older versions of Savant. If it does happen, please file a support request explaining exactly what steps you were performing when the error occurred so any possible issues can be resolved.

## Acceptance Field Truncated

Savant encountered an improperly formatted request – the field of the request that contains information about what type of media a client prefers to receive was incorrect. Savant uses this header to determine which form a file to send a client. When this field is corrupted, Savant will use its default file types, which should work properly with most clients.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## **Accept-Language Field Truncated**

Savant encountered an improperly formatted request – the field of the request that contains information about a client's preferred language was incorrect. Savant uses this header to send the proper language version of a file to a client. When this field is corrupted, Savant will send the English-format file by default if multiple language types exist.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## Authorization Field Truncated

Savant encountered an improperly formatted request – the field of the request that contains information about a client's access authorization was corrupted. Savant uses this header to limit access to server resources and files. When this field is incorrect, Savant will deny access to the restricted resource or file.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation. If this error occurs frequently, it's possible that a hacker is attempting to access files on your system.



## Buffer Overflow In Image Map

Savant tried to read a line from a server-side image map file that was too long. A line that overflows the buffer is usually indicative of an error in the image map file.

Savant will still properly display the page that contains the faulty image map, but it will not respond to clicks in the “hot spots” of the image map.

To solve this problem, check all of your image map files for an unusually long line. When you find it, check that the line follows proper syntax conventions for server-side image maps.

## Connection Field Truncated

Savant encountered an improperly formatted request – the field of the request that contains instructions on how to deal with connections by proxy servers is incorrect. Savant uses this header to manage access by proxy servers. When this field is corrupted, Savant will close proxy connections as soon as a client connection is terminated.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## Content-Encoding Field Truncated

Savant encountered an improperly formatted request – the field of the request that specifies the encoding scheme used by a requested file was incorrect. Savant only uses this field for compliance with the HTTP 1.1 specification and otherwise ignores it. When this field is corrupted, Savant will service the request normally.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## Content-Length Field Truncated

Savant encountered an improperly formatted request – the field of the request contains the length of the requested file was incorrect. Since some resources are dynamic in nature, this field is often omitted or ignored. When this field is corrupted, Savant will service the request normally.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## Content-Type Field Truncated

Savant encountered an improperly formatted request – the field of the request that specifies the media type of the requested file was incorrect. When this field is corrupted, Savant sends the requested file using its default media type and description.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## Couldn't Convert File Times

Savant was unable to convert the format of a file's time stamp. The Internet standards for file times are different from the standards for the Windows file system, so the time stamps must be converted between the different formats.

A file that has a corrupted time stamp causes this error. Usually a file's time stamp is corrupted by a virus or a file system fault. Savant will still serve the file, but some browsers and applications may not accept the file.

To correct this error, open the file with an invalid time stamp in the appropriate application and save it. This should replace the corrupted time stamp with a valid one.

## Date Field Truncated

Savant encountered an improperly formatted request – the field of the request that contains the request's time and date was incorrect. Savant uses this field to serve requests in the proper order; a request without a proper time and date will time-out and not be served. When this field is corrupted, Savant will usually ignore the request.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## Error Closing Registry Key

Windows returned an error when Savant tried to close a registry key. Savant closes a registry key after it reads configuration information from the registry.

The volatile nature of the Windows registry is usually to blame for this error. In most cases, Savant will continue serving normally after this error occurs. On rare occasions Savant ignore some requests, or even stop servicing requests all together.

While Savant will more than likely continue to normally serve requests, you should probably restart your computer upon receipt of this error.



## **Failure To Clean Up WinSock**

Savant was unable to properly shut down WinSock while shutting down. Since this error only occurs when Savant is shutting down, it ignores the error and continues the shut down process.

This error is usually caused by a TCP/IP fault or an operating system problem that has affected WinSock. Savant will shut down normally, but its TCP/IP structures will remain resident, preventing any web server from running on your system until it is restarted.

To solve this problem, restart your computer.

## Failure To Convert ASCTime Time String

Savant was unable to convert the format of a file's time stamp. The Internet standards for file times are different from the standards for the Windows file system, so the time stamps must be converted between the different formats. Windows file systems store file dates in a format known as ASCTime, which must be converted to the Internet standards of RFC 822 or RFC 850 before it can be used.

A file that has a corrupted time stamp causes this error. Usually a file's time stamp is corrupted by a virus or a file system fault. Savant will still serve the file, but some browsers and applications may not accept the file.

To correct this error, open the file with an invalid time stamp in the appropriate application and save it. This should replace the corrupted time stamp with a valid one. If this doesn't work, restart your computer.

## **Failure To Convert RFC 822 Time String**

Savant was unable to convert the format of the time stamp included in a file request from a client. The Internet standards for file times are different from the standards for the Windows file system, so the time stamps must be converted between the different formats. The RFC 822 format must be converted to the Windows ASCTime format for Savant's advanced serving features to work.

This error is caused by the client, which sent a corrupted or improperly formatted request to Savant. Savant will serve the file, but will utilize the older HTTP 1.0 protocol instead of the more efficient HTTP 1.1 protocol.

You can safely ignore this error – it has no effect on Savant's operation.

## **Failure To Convert RFC 850 Time String**

Savant was unable to convert the format of the time stamp included in a file request from a client. The Internet standards for file times are different from the standards for the Windows file system, so the time stamps must be converted between the different formats. The RFC 822 format must be converted to the Windows ASCTime format for Savant's advanced serving features to work.

This error is caused by the client, which sent a corrupted or improperly formatted request to Savant. Savant will serve the file, but will utilize the older HTTP 1.0 protocol instead of the more efficient HTTP 1.1 protocol.

You can safely ignore this error – it has no effect on Savant's operation.

## Failure To Create CGI Process

Savant could not create a process to run a Common Gateway Interface application. When this error occurs, an HTML-formatted error message is sent to the client, and Savant continues to service file requests normally.

This error usually has its roots in either a lack of system resources or a misconfiguration of Savant.

To solve this problem, try the following steps:

1. Verify that you have enough memory for the CGI program or script to be executed.
2. If the CGI application is a script, verify that you have the proper interpreter (usually the Perl 5 interpreter) installed on the server.
3. Savant uses the Windows file manager to determine which interpreter to execute a script with. This error will result if the script has an extension that isn't associated with a specific interpreter in the Windows file system. For example, a Perl script with the extension .pl must be associated with the perl.exe interpreter. By default, Savant 3.0 sets up the proper associations for Perl.

## Failure To Create CGI StdErr Pipe

Savant was unable to set up the data pipe required by a Common Gateway Interface application to transmit errors back to the client. Savant will send an HTML-formatted error message to the client and continue to serve file requests normally.

This error results when Savant cannot access the directory set as the CGI temporary directory.

To solve this problem, try the following steps:

1. Using the Savant configuration utility, verify that the directory being used by Savant for temporary files exists.
2. Verify that you have sufficient memory and disk space for Savant to create data pipes.
3. Restart your computer to clear any possible file locks.

## Failure To Create CGI StdIn Pipe

Savant was unable to set up the data pipe required by a Common Gateway Interface application to transfer data from the client to the CGI application. Savant will send an HTML-formatted error message to the client and continue to serve file requests normally.

This error results when Savant cannot access the directory set as the CGI temporary directory.

To solve this problem, try the following steps:

1. Using the Savant configuration utility, verify that the directory being used by Savant for temporary files exists.
2. Verify that you have sufficient memory and disk space for Savant to create data pipes.
3. Restart your computer to clear any possible file locks.

## Failure To Create Directory Listing

Savant was unable to create the HTML page that contains a file listing for a directory. When this error occurs the client will receive an HTTP 404 error, and Savant will continue to serve file requests normally.

This error results when Savant is unable to dynamically create an HTML format directory listing, usually because the directory doesn't exist or the Windows file system has been corrupted.

To solve this problem, try the following steps:

1. Use the configuration utility to verify that the requested directory exists and is properly mapped.
2. Restart the computer to clear any possible file locks.



## **File I/O Error**

Savant encountered a problem while trying to read or write a file. Savant will terminate the process that generated the fault, and continue to service other requests normally. Since the process was terminated immediately upon receipt of the error, the client's browser will time out and the request will have to be repeated.

This error usually results from a caching or locking fault in the Windows file system.

Normally you can ignore this error. If it begins to occur on a frequent basis, you should restart the computer to eliminate any possible file lock or cache faults. If this error continues even after system restart, computer virus activity should be checked for.

## **From Field Truncated**

Savant encountered an improperly formatted request – the field of the request that contains the client's e-mail address was incorrect. This is an optional field that some Common Gateway Interface programs use. When this field is corrupted, Savant will return a null value to any CGI program that requests the client's e-mail address.

This error is caused by the client browser, not Savant. Usually this error is produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## **If-Modified-Since Field Truncated**

Savant encountered an improperly formatted request – the field of the request that contains information about the last time a file was modified was incorrect. Savant uses this header in conjunction with the HTTP 1.1 protocol to optimize client-side caching. When this field is corrupted, Savant will send the file by default and ignore any possible client-side caching.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## **Invalid Client Socket**

Savant attempted to respond to a client's request, but the client's socket interface has been corrupted or the client's request was improperly formatted. Upon receipt of this error, Savant ignores the request and continues to service other file requests normally.

This error is usually caused by the client computer crashing or losing its network connection. You can safely ignore this error – it has no effect upon Savant's operation.

## Invalid Or Missing File Time

Savant was unable to parse a file's time and date stamp. Savant will continue to try to serve the request normally.

This error is caused by a corrupted file time or date. To correct this error, open the file with an invalid time stamp in the appropriate application and save it. This should replace the corrupted time stamp with a valid one.

## **MIME Version Field Truncated**

Savant encountered an improperly formatted request – the field of the request that contains information about the MIME version was incorrect. When this error occurs, Savant assumes the MIME version to be 1.0.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## **Parse Error In Image Map *File Line# Error***

Savant encountered an error while attempting to parse a server-side image map file. Savant will ignore the line of the image map it was unable to parse.

This error is caused by a line of the image map file violating proper syntax or being poorly formatted. To solve this problem, carefully check the line number indicated by the error message for the error indicated by the message.

## Pragma Field Truncated

Savant encountered an improperly formatted request – the field of the request that contains optional information was incorrect. Since this field is optional, Savant will ignore it and try to complete the file transfer.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.



## Referrer Field Truncated

Savant encountered an improperly formatted request – the field of the request that identifies the HTML document linking to the requested file was incorrect. This field of the request is non-essential, so Savant will ignore it and try to complete the file transfer.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## **TCP Buffer Overflow**

One of the buffers used by Savant for a data transmission purpose has overflowed. Savant will terminate the process that generates the overflow, and continue to service other requests normally.

This error is caused by a TCP fault, and can be safely ignored. The client that made the request will time out and re-transmit the request. If the error begins to occur on a frequent basis, restart the computer to clear any faults.

## Unexpected Region Value In Image Map

Savant encountered an erroneous constant for a region value in a server-side image map. Savant will complete the file request, but will ignore mouse clicks in the “hotspot” defined by the erroneous value.

This error is caused by an incorrect value in an image map definition file. To solve this problem, check the server-side image map file for an out of range value.

## Unknown Message

A TCP/IP message was sent to Savant's listening socket that Savant could not interpret. Savant ignores the message and continues serving file requests normally.

This error can be safely ignored – it has no effect on Savant's operation. Usually this error's source can be traced to a client's application trying to access a server on TCP port 80 without using the HTTP protocol. If this error occurs on a regular basis, there's a possibility that someone is trying to gain unauthorized access to files on your web site.

## User Agent Field Truncated

Savant encountered an improperly formatted request – the field of the request that identifies the type of browser the client is using was incorrect. This field of the request is non-essential, so Savant will ignore it and try to complete the file transfer.

This error is caused by the client browser, not Savant. Usually this error is only produced by a browser that has been corrupted and needs to be re-installed. You can safely ignore the error – it has no effect upon Savant's operation.

## **400 – HTTP Version Not Supported**

A client made a request in an HTTP format not supported by Savant. As of the beginning of 2001, Savant 3.0 supports every standard version of the HTTP protocol. Receipt of this error indicates that the client sent an HTTP request that was improperly formatted.

## **403 – Forbidden**

A client made a request to a restricted file or resource, and did not authenticate itself properly. The request was denied, and an HTML-formatted error message was sent to the client.

## **404 – File Not Found**

A client requested a file or resource that does not exist. An HTML-formatted error message was sent to the client.



## **405 – Method Not Supported**

The client attempted to request a resource using a request method not supported by either that resource or Savant. Some client programs (such as Netscape's roaming access applications and some of Microsoft's web-enabled applications) use proprietary request methods that are only supported by proprietary servers. Savant strictly supports the official HTTP 0.9, 1.0, and 1.1 protocols and does not support these proprietary request methods.

## **500 – Internal Server Error**

A server process (such as a Common Gateway Interface application or a servlet) crashed or was unable to function properly. An HTML-formatted error with extended error information is sent to the client.

## **501 – Internal Server Error**

A client requested an operation that cannot be performed by the server for security or implementation reasons. An HTML-formatted error message with extended error information is sent to the client.

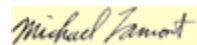
# Welcome

Welcome to Savant 3.0, the latest version of the Savant web server. I started programming Savant in the Fall of 1997 as a small web server for my personal use. I just wanted to put up a simple HTML page at the time, and made the mistake of trying to use Microsoft IIS. IIS is a decent web server, but to put it bluntly, it's a pain in the rear to get working properly. It's so loaded with features and switches that you can never find what you want. IIS is also full of security holes that clever people keep discovering, so running IIS turns into a cat-and-mouse game between the server administrator and those bent on mischief. Microsoft's Personal Web Server wasn't much better – at the time, it didn't even let you serve documents anywhere but your local network. Newer versions do, but they suffer from the same problems as IIS. I tried the freeware and shareware programs on Tucows and Cnet, but couldn't find anything that I really liked. Being more than a little frustrated at the time, I decided to write my own. (The fact that I was using this as an excuse to procrastinate doing my computer science homework didn't hurt.) And so Savant was born.

At first, Savant was just a programming toy for me – I'd work on it for a few hours on the weekend when I didn't have anything better to do. I picked the name "Savant" for some odd reason that I don't remember – the word means "wise". Unfortunately, my first few attempts were a little less than Savant – an early one randomly sprayed packets onto the network as fast as it could, creating some interesting network problems one Saturday morning. Another attempt served files a little too well – it served my entire hard drive instead of just the directories it was supposed to. But after almost a year of programming and discovering every way possible to crash Windows 95 (and there are a lot, trust me), the first version of Savant was done in the summer of 1998.

Several years and revisions later, we have Savant 3.0 here before us. This version includes a real-time monitoring console, support for ISAPI applications and WinCGI, better support for CGI, and numerous efficiency and bug fixes.

I'd like to thank you for choosing Savant as your web server. I hope you enjoy using Savant as much as I enjoyed writing it.



Michael Lamont  
February 23, 2001

## General Information

Savant is a full-featured open source / freeware web server designed to be run under Microsoft Windows 95/98/ME/NT/2000. Savant was designed to be easy to use first, and fast and secure second.

Savant will run on almost any computer capable of running Windows 95 or better, but the minimal requirements are:

**Operating System:** Windows 95 or higher..

**Processor:** Intel or compatible 486 running at 25 MHz as the minimum, Pentium or higher recommended.

**Memory (RAM):** 8 Megabytes minimum, 32 or higher recommended.

**Free Hard Drive Space:** Around 10 Megabytes, depending on how large you wish to make your web site.

# Features

Savant 3.0 includes these basic features, plus many more!

- HTTP/1.1 with keep-alive support
- CERN/NCSA Common Log Format logging, including browser and referrer entries
- Performance logs and error logs
- Configurable logging
- Server-side image maps
- CGI/1.1
- WinCGI
- ISAPI Application support
- User-based password protection
- Limit access by IP addresses
- Easy install/uninstall
- Graphical Configuration
- User-defined MIME Types
- Aliasing & file system mapping
- Simple directory listings
- Optional reverse DNS lookups
- Process control
- Perl for Win32 Included
- Monitoring Console
- Online help
- Realtime status display
- Complete source code for server and configuration utility available
- Web-based support system
- Support mailing lists

# Warranty & License

## **GNU GENERAL PUBLIC LICENSE**

*Version 2, June 1991*

Copyright ©1989, 1991 Free Software Foundation, Inc.  
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

### **Preamble**

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software - to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

### **GNU General Public License**

#### **Terms and Conditions for Copying, Distribution and Modification**

**0.** This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is

included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

**1.** You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

**2.** You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

**3.** You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software



interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

**4.** You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

**5.** You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

**6.** Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

**7.** If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the

free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

**8.** If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

**9.** The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

**10.** If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

## **NO WARRANTY**

**11.** BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

**12.** IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

# Credits & Acknowledgements

## **Programming**

Michael Lamont

## **Beta Testing**

Jason Grace

David Wilkins

## **Special Acknowledgements**

Maranda & Yonathan, brix & Homie,

## For The Impatient

This section is designed as a quick guide for those people who just can't wait to start serving their web content and don't want to bother to read the entire help file. Follow these steps:

1. Make sure Savant is running (the stylized 'S' icon in a blue box is in the system tray).
2. Start Windows Explorer, and find where you installed Savant on your system (unless you changed it during the install process, it's at **C:\Savant**).
3. In the **root** subdirectory of **C:\Savant**, place your HTML and image files. Make sure that the default file you want users to see when they connect to your site is named **index.html**
4. Your very own web site is ready to go!

## Server Core

The server core is the heart of Savant – it's the actual program that receives and answers requests from web browsers. The server core is much like the core of a nuclear reactor – it does the actual work, while other systems deal with the details of operation. Savant's core reads its startup parameters from the Windows registry, and outputs the results of operations to log files and the control console.

## Configuration Utility

The graphical configuration utility is used to change the way the server core operates. The utility is designed to be intuitive and easy to use so you don't have to spend hours configuring the server. There are two key points to remember about the configuration utility:

1. While the utility is running, the server core shuts down to prevent data corruption. In short, your web site will be inaccessible while the configuration utility is running.
2. If your web site is currently active, the configuration utility won't start until the last pending connection has been serviced. Sometime this can take as much as 5 minutes due to network vagaries – don't be surprised if nothing happens for a few minutes after you click on an option to start the configuration utility.

For more information about each of the property pages in the configuration utility, see the section of the help file labeled "Configuring Savant".

## Control Console

The control console is the primary method of keeping track of what Savant is doing, and changing it to fit your needs. The console displays every request made to Savant in real time, so you can see which users are accessing your web site, and what pages they are viewing. In addition, the control console has menu buttons that provide access to this help facility, the configuration utility, and shutting down the server.

## How Do I Replace The Default Page With My Own Stuff?

To replace the default page with your own web content, make sure that the first web page you want users to see when they connect to your site is named **index.html**, or the filename that you specified in the [HTTP property sheet](#). Next, copy your HTML files, images, and other web content to the **root** subdirectory of the Savant installation, usually located at **C:\Savant**.

Any subdirectories you make in the **root** directory can be accessed as subdirectories from the web through Savant – for example, if you use Windows Explorer to create a subdirectory of **root** named **test**, you can access it in a web browser by putting in `http://localhost/test`. If you wish to create directories that are accessible from the web in locations other than the **root** directory create them normally, then use the [Paths property sheet](#) to make them web accessible.



## How Do I Set Up CGI Scripts?

To use a CGI script or program with your web site, place it in the **cgi-bin** subdirectory of the Savant directory, usually located at **C:\Savant**. Reference the CGI script or program using HTML forms, or a direct reference if the CGI generates an image that you wish to include in your web page.

Note that Savant only sets up the .pl extension for use with the Perl interpreter. If you want Perl to interpret files that end in the .cgi extension, you need to associate that extension with the Perl interpreter in Windows. The easiest way to do this? Right-click on a file with the .cgi extension, and select the "Open With" option from the context menu that appears. Make sure the "Always use this program to open this file" checkbox is checked, and select the Perl interpreter from the list of programs. Click the OK button, and ignore the funky error that appears. The .cgi extension is now associated with the Perl interpreter.

## How Do I Get Savant Source Code?

In the past, I released Savant as a freeware tool but held back the source code. My logic for this was that it was possible to perform a nasty Denial-of-Service attack by slightly altering the source code (I discovered this by accident one Saturday morning.) I've always been a big supporter of the Open Source movement, but I'm an even bigger supporter of network and system security.

But with the recent advent of automated DoS tools such as Trin00 and Stacheldrucht (which were used to take down several popular web sites in mid-2000) it no longer made any sense to hold back the Savant source code for that reason. I usually receive an email every week or two from an aspiring programmer asking to see the Savant source to use as a model, and it makes no sense to keep the information from them. I've often gone through hundreds of pages of open source code to find the solution to one of my own programming dilemmas, and if I didn't open up the Savant source code that would essentially amount to taking from the community without returning anything.

So I proudly announce that Savant 3.0 is completely open sourced, and available for perusal and modification under the terms of the GNU General Public License. A current copy of the source code will always be available from <http://savant.sourceforge.net>, as well as being stored in the SourceForge CVS repository at [cvs.savant.sourceforge.net](http://cvs.savant.sourceforge.net)

## Contacting the Author By E-mail

If you have comments or suggestions, you can reach me at [lamonml@hera.wku.edu](mailto:lamonml@hera.wku.edu)

I receive so many e-mail messages a day dealing with Savant that it may take me a few days to get back to you – please be patient.

The SourceForge system provides several automated tools to request support and report bugs. I'd really appreciate it if you used those tools rather than contacting me directly, since they really speed up the support process. If you submit your support request or bug using the SourceForge system, I'll almost always have a response to you within 15 days. If you send it to me by email, it can sometimes take several weeks for me to get around to it. You can access these tools by visiting <http://savant.sourceforge.net> with any web browser.

If 15 days is too slow for you or you'd like to help other people out, I highly suggest joining one or more of the Savant mailing list. You can check them out at the above web URL.

## **Contacting the Author By Snail Mail**

If you wish to mail a letter or package (no letter-bombs please!) to the author, you can send it to the following address:

Michael Lamont  
Rodes-Harlin #902  
1474 Kentucky ST  
Bowling Green, KY  
42101,USA

## **Savant On The Web**

Information, updates, source code, and patches for Savant can be found at the Savant website,

<http://savant.sourceforge.net>

