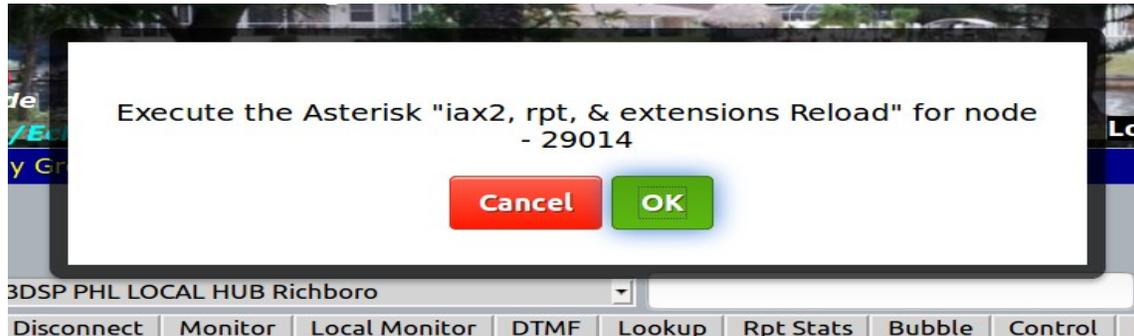


Supermon2 V2.0 Changes

Version 2.0 adds significant changes to Supermon2 making it easy to add security when used with multiple users and in general to customize your systems. For mutli-user logins there are two key changes individual button authorization by user and initialization (ini) file selection by user. Below is a description of how each works. Both the favorites and control panel are also user configurable.

Another change to make it fancier is action results are displayed graphically. Here is an example:



The directory structure has been changed. **ALL** user configurable files have been moved to the “**user_files**” directory under the supermon2 directory. There are **NO** user configurable files in the supermon2 directory. Files in the “**user_files**” directory are:

- global.inc** - user configurable options
- controlpanel.ini** - control panel configuration
- favorites.ini** - favorites panel configuration
- authini.inc** - user to ini file mappings
- authusers.inc** - user to button mappings
- allmon.ini** - default master ini filename
- nolog.ini** - ini files used when configured and not logged in
- set_passwd.sh** - Script to set Supermon2 passwords
- Example Files** - Many example files
- README file** - a readme file explaining the files
- background.jpg** - Header background image file
- favnolog.ini** - Favorites ini for non logged users
- cntrlnolog.ini** - Control Panel ini for non logged users

Several other files could be created in the “**user_files**” directory by the user in implementing the user/ini mappings. For instance if you had a user “**sam**” and you wanted to give that user their own customized ini file you could call it “**sam-allmon.ini**” and map it to the user “**sam**” in the **authini.inc** file. More on this later.

In addition there are several example files and backup files provided. It is important to keep backups and also to preserve the original backup files in case you may need them in the future. Example files should never be changed. Files in the “**user_files**” directory are protected and will not be overwritten in future updates but example files may be updated.

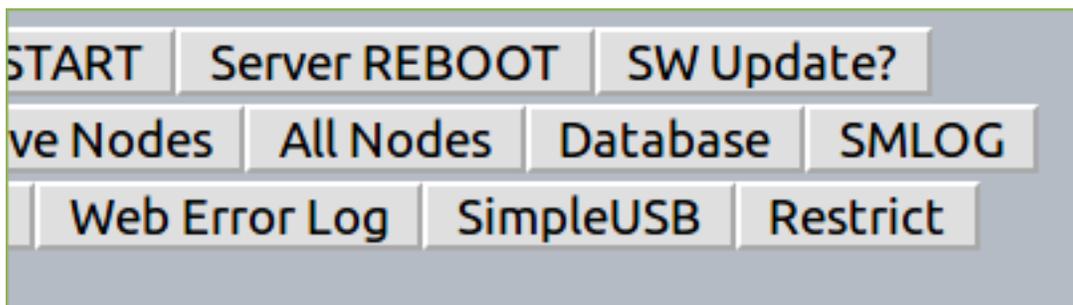
There is also a new script file in the “**user_files**” directory called **set_password.sh**. This script reads, creates, deletes, or updates the password file for Supermon2. It is an easy way to manage your passwords.

Password management is always important but even more so with multiple users. Using this script makes that task easier.

VERY IMPORTANT is that the above files **authusers.inc** and **authini.inc** at distribution are named **Xauthusers.inc** and **Xauthini.inc**. This eliminates the use of the button/user and ini/user mappings making Supermon2 behave as it did in the past using one ini file named **allmon.ini** and every logged in user has all buttons capability. To implement either of these features the preceding “X” must be removed from the filename but before doing that read and understand how both of these options work. Not everyone would want or need those features but if you do it will come in quite handy. Here are some scenarios where you might use them.

The user/button mappings allow you to setup multiple users or user groups that see and can use different buttons when they login. A typical use of this would be a hub when you advertise the url for your Supermon2 page. You would be the **admin** user and have all button privileges. You might have several “sysops” watching over your hub or repeater and they would have most but not all buttons available. As an example only the **admin** user could use the configuration editor, GPIO, system reset or reboot buttons or anything else you did not want out of your control. You could have multiple general user levels that would have less capability and finally the not logged in user would have no buttons except the display configuration. This is all configured in the **authusers.inc** file in the **user_files** directory. Each button is the key of an array where each user name is stored. If a name appears it is authorized for that button and it appears when they login. More detail on this is shown below.

The user/ini mappings allow you to tailor what nodes and links are visible on a per user basis. This is also useful if you don't have actual users other than yourself and you want to customize multiple instances of Supermon2. You could have multiple tabs open to the same Supermon2 but be logged into different users names for each and each would be mapped to a different ini file. Lets say you managed many nodes and you wanted to quickly display different combinations. You would just log into different users on multiple tabs of your browser. Each user would be mapped to a different ini file in the **authini.inc** file in the **user_files** directory.



Two new buttons have been added, SimpleUSB moved from configuration editor and SMLOG showing Supermon2 login/out info. Both are programmable by user name.

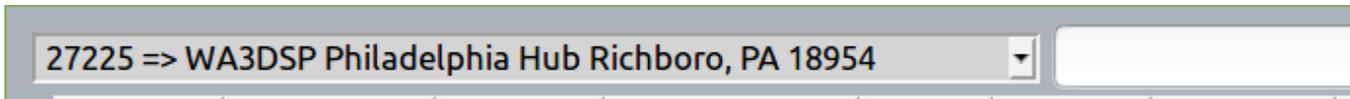
Another very significant change allows the RELOAD, AST START, AST STOP, RESTART, Rpt Stats, SimpleUSB, Database, and Restrict buttons to control any valid node in your current ini file that is connected via the manager. This means you have significant control over all your nodes in one place in Supermon2 wherever they are in the world. This is in addition to the current multi-node control capability with Connect, Disconnect, Monitor, Local Monitor, and DTMF.

For convenience the Restrict button carries the node number into the node block so simply selecting the

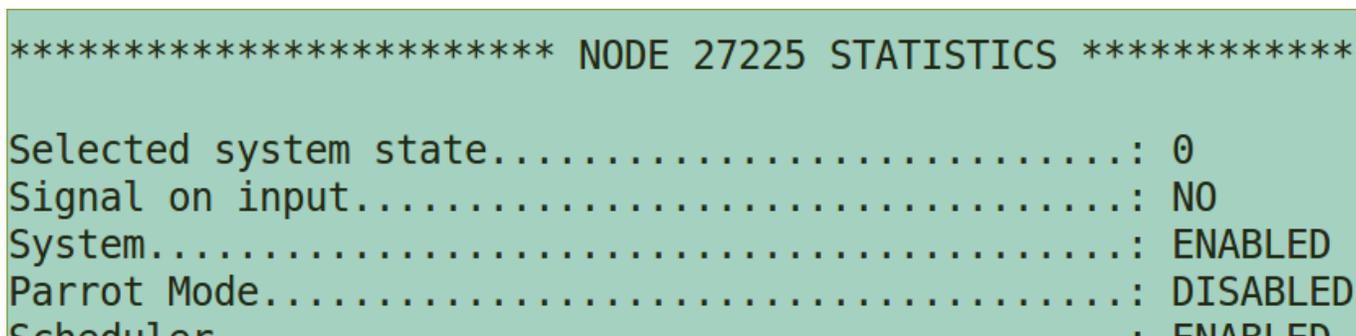
node brings both the node number and the server node number into Restrict making it very easy to add or delete. Also system info has more information, the logout button shows the user name, and the base of the display shows current login and ini file information.



Since many of the options work on any node specified in the current ini file and manage connected the node number of the server you are sending the command on is displayed. Here are a few examples:

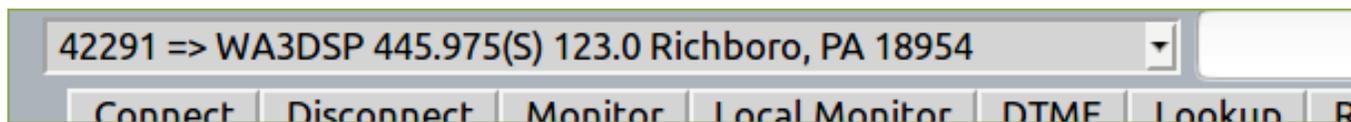


This is an example of RptStats – whatever node you select in the dropdown is displayed.

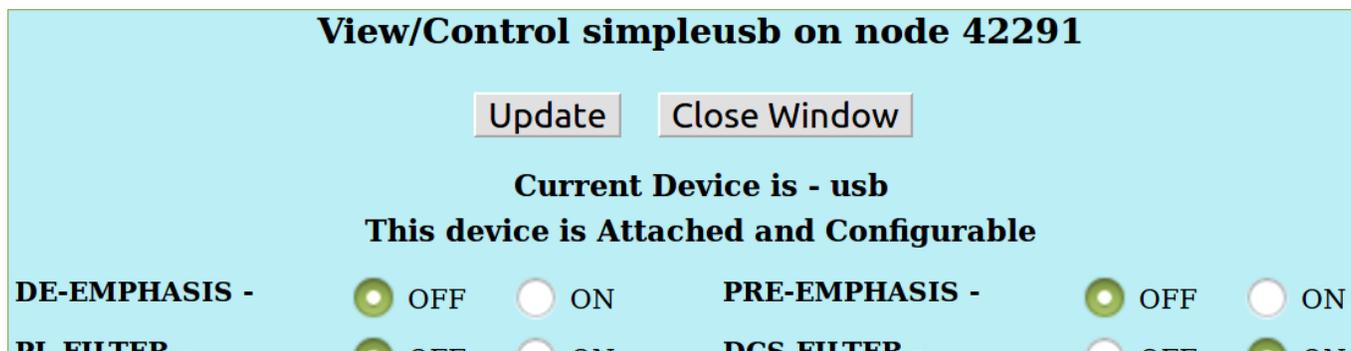


With the node title in the corresponding display and results from that node

Another example for simple usb control. Any node in the ini file that has simpleusb active can be controlled.



Here I select another node not on the Supermon2 server then the SimpleUSB button . It could be anywhere reachable by Internet. I would then be able to change simpleusb settings at that node.



So let's detail how to configure user/button and user/ini mappings.

Button Authorization by User

When implemented this presents to each login user of Supermon2 only the buttons authorized for that user. To turn this capability on you must first rename the **user_files/Xauthusers.inc** removing the preceding "X" and then configure the **authusers.inc** file. It is important that this file be configured properly. Misconfiguration could stop Supermon2 from running properly or running at all. In the event you have a problem you cannot resolve configuring this file you can always revert by renaming the file with a beginning "X" to disable user/button configuration and returning to all users have all buttons.

ALL user configurable files in the "/srv/http/supermon2/user_files" directory

Logged in as - '**admin**' using ini file - '**user_files/allmon.ini**
Supermon2 Logged OUT ini - "**user_files/allmon.ini**"

Selective ini - **INACTIVE** - Using just allmon.ini file
Button selective based on username - **ACTIVE**

Both the "selective ini/user" and "selective button/user" are indicated as active or inactive in the system info display.

authusers.inc Setup

The **/srv/http/supermon2/authusers.inc** file is a list of variable arrays defining each button. Each variable has a comment line before it describing what button it controls. The users are defined in arrays. You can have as many users per button as you desire. The example file has user names admin and user1 through user4. You could assign different users to each name with a password. In this case the password for each level would have to be the same. Alternatively if you want to assign multiple passwords per level you could name the levels further like **admin-sam** or **user2-joe**, or **admin1a,b,c,d,e** etc. Any valid user name is possible. This is just an example but you can only have one password assigned to a specific user name. So you could not have two bills, or joes unless they both shared the same password.

The **authusers.inc** file by default looks like this -

```
// Button Authorization by user name
// DO NOT change variable name only
// user name(s) in quotes ending with a semicolon
// Multiple login names per item can be defined
// EXAMPLE- $PERMUSER=array("admin1","admin2","joe","sam");
// Names must exactly match login names including case.
// YOU MUST HAVE THE SYNTAX CORRECT IN THIS FILE
// OR SUPERMON WILL NOT RUN!!!
// NOTE - Items marked with (*) are considered
// a security risk to make public others at your discretion.
```

```
//
// Use the nano editor to mass change names in this file using
// the "control \" option. This will allow you to change the
// below usernames to match real usernames you have created on
// your Supermon2 server. Then edit each button to add or change
// authorization for each user.
//
// Begin configuration area
//
// Perm checkbox
$PERMUSER=array("admin","user1","user2","user3","user4");

// Connect button
$CONNECTUSER=array("admin","user1","user2","user3","user4");

// Disconnect button
$DISCUSER=array("admin","user1","user2","user3","user4");

// Monitor button
$MONUSER=array("admin","user1","user2","user3","user4");
```

... and many other button definitions.

As shown **ALL** buttons are defined to user “**admin**” or well as "**user1**", "**user2**", "**user3**", "**user4**". This means that unless you create an “**admin**” or one of the “**userx**” user and password and log in as that user you will see no buttons. For many users who would be the only login to their supermon2 you could either leave the **authusers.inc** file disabled by keeping or adding an “**X**” at the beginning of the name or create an “**admin**” user/pass with the **authusers.inc** file implemented. Either way you would see all the buttons and have all Supermon2 capability.

For those that want to customize logins you edit the **user_files/authusers.inc** file and change the user settings per button to the configuration you desire.

The default **authusers.inc** file has five levels. You are not limited to the five but for most users that might be adequate. Five levels is actually six including not being logged in which is no buttons. The default logins are **admin**, **user1**, **user2**, **user3**, **user4**. So **admin** would be all buttons (defined in every button variable) and user1 through user4 would be customized to whatever buttons each users would see. So here is an example of how you could edit the **authusers.inc** file. First of all **admin** is defined for every key. You would probably want to keep **admin** to yourself. **user1 through 4** would then be customized for different levels of users. Most general users only concerned with who is connected would not be logged in at all and see no buttons. They would also use the **nolog.ini** file if that was setup as described below.

So here is an example **authusers.inc** using these five tiers. Rather than deleting a user name where the button is not available for that user I simple put an “**X**” before the user name. This makes it easy to change in the future if you want to turn the button for that user back on by just removing the “**X**”. Note in the file some buttons are marked with an “*****” to indicated it is not a good idea to give this feature to anyone but the **admin** user. There certainly may be others you want to limit but these could be particularly harmful to allow just anyone to use.

Example authusers.inc file -

<?php

```
// Button Authorization by user name
// DO NOT change variable name only
// user name(s) in quotes ending with a semicolon
// Multiple login names per item can be defined
// EXAMPLE- $PERMUSER=array("admin1","admin2","joe","sam");
// Names must exactly match login names including case.
// YOU MUST HAVE THE SYNTAX CORRECT IN THIS FILE
// OR SUPERMON WILL NOT RUN!!!
// NOTE - Items marked with (*) are considered
// a security risk to make public others at your discretion.
//
// Use the nano editor to mass change names in this file using
// the "control \" option. This will allow you to change the
// below usernames to match real usernames you have created on
// your Supermon2 server. Then edit each button to add or change
// authorization for each user.
//
// Begin configuration area
//
// Perm checkbox
$PERMUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Connect button
$CONNECTUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Disconnect button
$DISCUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Monitor button
$MONUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Local Monitor button
$LMONUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// DTMF button (*)
$DTMFUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Asterisk lookup button
$ASTLKUSER=array("admin","user1","user2","user3","user4");

// RPT stats button
$RSTATUSER=array("admin","user1","user2","user3","user4");

// Bubble button
$BUBLUSER=array("admin","user1","user2","user3","user4");
```

```
// Favorites button
$FAVUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");
// Control Panel button (*)
$CTRLUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Configuration Editor button (*)
$CFGEDUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Asterisk Reload Button (*)
$ASTRELUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Asterisk start button (*)
$ASTSTRUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Asterisk stop button (*)
$ASTSTPUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Asterisk restart button (*)
$FSTRESUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Asterisk reboot button (*)
$RBTUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Hamvoip update check button
$UPDUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Hamvoip howto button
$HWTOUSER=array("admin","user1","user2","user3","user4");

// Allstar wiki button
$WIKIUSER=array("admin","user1","user2","user3","user4");

// CPU status button (*)
$CSTATUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Allstar status button
$ASTATUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Allstar node database button
// Only active if using file node database
$EXNUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Node info button
$NINFUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Active node button
$ACTNUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// All nodes button
```

```

$ALLUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");
// Supermon2 Log Display button
$SMLOGUSER=array("admin","user1","user2","user3","user4");

// Database button (*)
// Note button only available if
// setup in global.inc
$DBTUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// GPIO button (*)
$GPIOUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Linux Log button
$LLOGUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Asterisk log button
$ASTLUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// Connection log button
$CLOGUSER=array("admin","user1","user2","user3","user4");

// IRLP LOG button
$IRLPLOGUSER=array("admin","user1","user2","user3","user4");

// Web access log button (*)
$WLOGUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Web error log button (*)
$WERRUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

// Restrict button (*)
$BANUSER=array("admin","user1","Xuser2","Xuser3","Xuser4");

// simpleusb button (*)
$SUSBUSER=array("admin","user1","user2","user3","user4");

// System Info button (*)
$SYSINFUSER=array("admin","Xuser1","Xuser2","Xuser3","Xuser4");

?>

```

If a user logs in as one of the five login user names they will be authorized the buttons where that user name is defined in the **user_files/authusers.inc** file. Buttons not defined for a specific user are **NOT** displayed. If you want to have multiple users at a specific level you could define them further and give each or a small group a different user name and password. Lets say you had an administrator group but you wanted to give out different passwords to each administrator. This would be useful in a situation where you might want to remove an administrator in the future. If they all had the same password then you would need to change the password for that user name and give it to all the currently authorized users of that name. On the other hand if you had say five administrators you could setup five user

name/password pairs and give one to each. So it could be something like admin1a, admin1b, admin1c, admin1d, and admin1e. How you actually name it and administer it would be up to you. It is very flexible. Using multiple user names and passwords and setting up the **authusers.inc** file with the names as levels like this would allow you to reassign a user name simply by changing the password. So if the person you assigned admin1c to was to be removed you would simply change the password and reserve that user/pass for another person.

To hopefully make it easier the default `user_files/authusers.inc` file has admin, user1, user2, User3, user4 defined for each button. You can mass edit names using the nano “control \” feature. You edit the file using nano and select what you want to change and then what you want to change it to. You could change all occurrences to something else or just some depending on what you want to do. There is also both an **authuser.inc.backup** file and **authuser.inc.example** file. Should something happen to the **authuser.inc** file you can recreate it from the backup file. You should always keep backups of the user files in the **supermon2/user_files** directory. Also always login as the users you create to make sure the buttons you authorized are correct. Remember in a pinch if something goes wrong with the **user/button** process you can always revert to the old “all logins get all buttons” by simply renaming the **authusers.inc** file to **Xauthusers.inc** but remember anyone that has login capability at that point will get ALL buttons.

Setting up the user/button authorization requires some configuration but once it is setup the Supermon2 sypop has total control over who and how your system is used.

authini.inc Setup

The `/srv/http/supermon2/user_files/authini.inc` file defines what ini file is used based on the login username. If this file does not exist (is renamed **Xauthini.inc**) the `user_files/allmon.ini` file is used for all logins. If the file exists and there is no login the file `user_files/nolog.ini` is used if it exists otherwise the `user_files/allmon.ini` file is used. If the username key exists then the associated ini filename is used otherwise `/user_files/allmon.ini` is used. Filenames are recommended to be the 'username-allmon.ini' to identify them as **allmon.ini** type files. All files are in the **supermon2/user_files** directory. Here is an example **authini.inc** file showing login user 'admin' using the **admin-allmon.ini** file, user 'doug' using the **doug-allmon.ini** file, user 'wa3dsp' using the **wa3dsp-allmon.ini** file, and user 'w3xyz' using the **admin-allmon.ini** file. If any of those files do not exist the default **allmon.ini** file is used. If no one is logged in and the **nolog.ini** file exists it is used otherwise **allmon.ini**.

```
<?php
```

```
$ININAME['admin'] = "admin-allmon.ini";  
$ININAME['doug'] = "doug-allmon.ini";  
$ININAME['wa3dsp'] = "wa3dsp-allmon.ini";  
$ININAME["w3xyz"] = "admin-allmon.ini";
```

```
?>
```

Users should ONLY change user names within quotes and brackets and associated file name within quotes. Format is important so **do not change anything else**. Assuming you want your users to have different ini files you should create as many user names as you have login users. Users can share ini files as shown above where user 'admin' and user 'w3xyz' both use the **admin-allmon.ini** file. In this case it could be a copy of the **allmon.ini** file. If this file is authorized and a user is not defined the **allmon.ini** file is used.

By default the **authini.inc** file has the single username '**admin**' created but commented out. It is recommended that the main control user of Supermon2 use the login name **admin**. If this is the case the example above shows the mapping of the **admin** user to the **admin-allmon.ini** file. You would copy the **allmon.ini** file to **admin-allmon.ini** and customize it if necessary. This file can then be edited by adding usernames and creating specific ini files for other users as desired.

PASSWORDS

Security is very important and even more so with the changes in Supermon2 . Please use good passwords!!! Do not reuse passwords. Browsers store passwords so the user should only need to enter it once per system. Use at least 10 characters of random upper/lower case letters, numbers, special characters. If you don't want to manually create a password use the **makepasswd** script available in all Hamvoip distributions. **makepasswd -h** for help **makepasswd -l10** for 10 character length. Don't use an exclamation point “!” in the password file.

The prime administrator should assign and be responsible for all passwords. People lose passwords so keep a written log of your users call/real name, the login name and password, and email. You can rest assured that a user will ask for a password they don't remember. Something like -

Doug, WA3DSP , admin, real-password, email-addr

There is no way to control a user from passing the password on to someone else that is not authorized so trust is important to know and understand. This is also a good reason to periodically change passwords and why the email address of the authorized users for notification is important.

Note the two tier password system using the edit directory has been eliminated in version 2.0!!! All passwords should be stored in the /srv/http/supermon2 directory. If **authusers.inc** is active and a button is not authorized it does not display. The script **set_password.sh** in the user_files directory will assist you in creating and maintaining your passwords.

Changing User names

You may feel a little overwhelmed at the complexity of the **authusers.inc** file with its many buttons and multiple user options. For the average user you can just use the either keep the **Xauthusers.inc** or remove the “X” and use it the way it is and make a new user name/password with user name “**admin**” and your selected password. When you log in you will see all the buttons. Should you want to use a different user name than **admin** you would then need to change all the “**admin**” values in **authusers.inc** to the new name. There is an easy way to do this in a mass fashion.

I created a script in **user_files** called **set_password.sh**. Use this script to setup your users and skip down to the “Mass Changing Usernames” section or if you would rather do it manually here is the method -

First delete the current password file.

```
cd /srv/http/supermon2  
rm .htpasswd
```

Then make a new user name/password with the new user name you want to use.

htpasswd -cB .htpasswd userID (subsequent additional entries use just -B)

Supply a password of your choice when prompted.<userID> is the login ID and can be any user name you would like to assign; your callsign or admin for example. DO NOT enter the < or >. Do NOT use a ! (exclamation point) in the password.

Mass Changing Usernames

Now lets say you want to change all the “**admin**” entries in the **authusers.inc** file to your new user name. This is easy to do in the nano editor.

cd /srv/http/supermon2/user_files

cp authusers.inc authusers.inc.bak - Make a backup copy

nano authusers.inc

Type- **Control **

It will ask- “**Search (to replace):**”

Enter- **admin**

It will ask- “**Replace with:**”

Enter- the new user name

Select “**A**” for **ALL**

And all the “admin” entries will be changed to the new user name.

Control x - to save, type [y]es and **return** and you are done

If you want to create additional logins repeat this process using the **set_password.sh** script or -

htpasswd -B .htpasswd userID

Then enter the user name to each button in **authusers.inc** authorized for that user. This is a tedious task but once it is done you should not need to do it again. Make backups of the **authusers.inc** and **authini.inc** files and be sure to keep records of your users and their passwords.

Control Panel and Favorites by User Name

Both the control panel and favorites windows are configurable by user name. By default this feature is inactive. It can be activated by renaming and removing the preceding 'X' in the user_files/Xfavini.inc and/or Xctrlini.inc files and by configuring these files with login user names. This is done in the same way as the authini.inc file is configured.

// EXAMPLES -

//\$FAVININAME['admin'] = "admin-favorites.ini";

//\$FAVININAME['doug'] = "doug-favorites.ini";

//\$FAVININAME['wa3dsp'] = "wa3dsp-favorites.ini";

// You should have at least one entry to facilitate changing your

// favorites login ini file name. Here it is shown as just mapping to

```
// the favorites.ini file but you can customize that. It is suggested
// that you name the files username-favorites.ini and they would always
// go in the supermon2/user_files directory.
```

```
// NOTE if the file user_files/favnolog.ini exists it will be used when
// there is no login to supermon2. If you want the nolog users to have
// the favorites.ini or some other ini simply copy it to user_files/favnolog.ini
```

```
$FAVININAME['admin'] = "admin-favorites.ini";
$FAVININAME['user1'] = "user1-favorites.ini";
```

And similarly for the Control Panel -

```
// EXAMPLES -
```

```
//$CNTRLININAME['admin'] = "admin-controlpanel.ini";
//$CNTRLININAME['doug'] = "doug-controlpanel.ini";
//$CNTRLININAME['wa3dsp'] = "wa3dsp-controlpanel.ini";
```

```
// You should have at least one entry to facilitate changing your
// favorites login ini file name. Here it is shown as just mapping to
// the favorites.ini file but you can customize that. It is suggested
// that you name the files username-favorites.ini and they would always
// go in the supermon2/user_files directory.
```

```
// NOTE if the file user_files/cntrlnolog.ini exists it will be used when
// there is no login to supermon2. If you want the nolog users to have
// the controlpanel.ini or some other ini simply copy it to user_files/cntrlnolog.ini
```

```
$CNTRLININAME['admin'] = "admin-controlpanel.ini";
$CNTRLININAME['user1'] = "user1-controlpanel.ini";
```

Selective Favorites ini - INACTIVE - using user_files/favorites.ini
Selective controlpanel ini - INACTIVE - using user_files/controlpanel.ini

The status for both favorites.ini and controlpanel.ini mapping is shown in the System Info screen. In the INACTIVE state the default files are used.

Auto Scroll to Top

To make it more convenient when executing buttons based on node numbers the page automatically scrolls to the top when you click on a node. The controlling node and action node are then displayed and a button selection which uses the node info like connect, disconnect, etc. can be selected. Also at the bottom of each controlled node when the number of nodes is display there is a ^^ selection which takes you to the top of the page as shown in the image below. This is particularly useful when you are controlling many nodes that use a lot of vertical page space.

| | | | | | | |
|--------------------|---|-------|-------------|----|----------|------------|
| 510500 | KB3JKB 445.5250 Simplex New Hope, PA | Never | ESTABLISHED | IN | 7:31:19 | Transceive |
| 542700 | K2SLB Mobile Node 434.85 Swedesboro, NJ | Never | ESTABLISHED | IN | 08:23:15 | Transceive |
| 31 nodes connected | | ^^ | | | | |