RhostMUSH Handbook

wstarter

Apr 19, 2021

TABLE OF CONTENTS

The RhostMUSH source tree offers many abilities and options not normally found in any other flavor of mush. This doesn't make it better than other servers (though we think so *j/k*) but it does give you a wider selection of configurability, which, as you know, is the best part of setting up a mush. (yea, right)

- A high-performance dual-quota system.
- A complete rewrite of key areas and referbishments of all the other areas.
- An on-line recover tool for accidental db destruction.
- Multi-wizard architecture for better control of staff.
- Multi-power system to tweek abilities of players.
- Multi-zone system where people can belong to multiple zones.
- Built in error correction for db corruption or other misuse.
- Built in accounting for those annoying twinks who try to hack.
- Many new functions and commands not seen anywhere else.
- A lot more that could drag this document out for pages.

CHAPTER

ONE

INSTALLING RHOSTMUSH

Table of Contents

1.1 RhostMUSH Requirements

1.1.1 System Requirements

- Operating System: Unix-like
  - Almost any variant of Unix or Linux should compile cleanly and run
  - Preliminary support for Windows with an equivalent development environment
- Memory: 12-100 MB
  - Depending on size of database and buffers, as well as selected options
  - 1 GB (memory and swap combined) is the minimum required to compile
- Storage: 100 MB and up
  - Depending on size of database and backup retention policy

Note: Third-party applications can use considerably more resources

1.1.2 Software Requirements

RhostMUSH is a Linux or Unix based server software that runs as a daemon on the host. In order to build this software, you will need the bare minimum of the ability to run 'make' commands.

Package requirements are as follows:

- bash/ksh/dash (or compatible shell - for use with build menu)
- glibc and gcc/clang (compiling the code)
- git (to clone the source and maintain patches)
- libcrypt (for password encryption - this is usually standard on unix based systems)

Optional Packages

RhostMUSH also offers optional linking and library attachments. For some of these libraries it will attempt to do auto-detection, but in a worse case scenario, there exists override hashes in the menu to disable options it thinks exist that do not.

Optional packages are as follows:

- libpcre (if you wish to use system pcre libraries instead of the built-in ones)
- mysql client & mysql_config (required for mysql capabilities)
- openssl dev libraries/headers (for MUX password compatibility, and digest() and advanced cryptology functionality.
- ruby/perl/python/etc (for custom interactive dynamic custom functions with the execscript() feature)
- sqlite3 libraries (required for sqlite capabilities)

1.1.3 Hosting Requirements

- You will need a stable host and access to open a single port number of your choice on the firewall.
- Most games choose a number between 1025 and 9999, by convention.
- Please make sure your debug_id matches the port number + 5.
  - So if your port is 4201, your debug_id will be 42015.
  - The debug_id is for use in the API daemon that runs Rhost as a container to keep track of heap, stack, and execution location.

1.2 Obtaining RhostMUSH Source Code

The only official source for obtaining RhostMUSH is through the 'RhostMUSH' github account. If the source code was obtained in some other manner, there are potentially any number of unintentional or intentional issues that you might run into.

The recommended method of obtaing RhostMUSH is to clone it's git reposistory:

```
git clone https://github.com/RhostMUSH/trunk Rhost
```

It is possible, but not recommended to download RhostMUSH via a web browser:

```
https://github.com/RhostMUSH/trunk/archive/master.zip
```

Note: This documentation generally assumes that you obtained the RhostMUSH source code by cloning it's git repository or at the very least downloading an archive of the source code from the GitHub website.

It also assumes that the base directory from which all commands are run and all files are looked for is that git repo's `Server/` directory, unless specifically noted otherwise.

1.3 Options for making a MUSH

There is a *lot* of options once your start making your MUSH, but there are also a few big choices right as you get started making a MUSH. No matter which choices you ultimately make, there are certain things you will need to know how to do. This chapter is going to walk you through making the simplest possible version of RhostMUSH. It's going to show you the things you would have to do no matter which choices you were making.

In the process of making that simplest possible RhostMUSH you'll also learn what the basics of those choices are and when and how you would make them. Later chapters will get much more into all of those choices, but they will also assume you know everything in this chapter already, or at least have it as a handy point of reference.

While there are lots of little options, there are three big areas where you make those choices.

1.3.1 Compile time options

One of the first steps of making a MUSH or really any computer program is to compile the source code. This takes what's basically text files full of code and turns them into a program you can run. Within that source code are a number of options to choose from, but any time you want to change one of them, you have to recompile the source code and then restart the MUSH.

These choices are typically made through the `confsource` menu which you'll be seeing momentarily. There are some pretty big choices here from whether or not you want hardcoded +help and comsys, deciding between a more secure server and certtain powerful but potentially dangerous MUSHcode options, and then whether or not you want to be able to connect to a variety of external programs like databases, webservers, and even other programming languages.

1.3.2 Configuration file options

While there are some options in RhostMUSH that can only be changed through recompiling the source code, there are way more options that can be changed without having to recompile. These choices are mostly made through the `netrhost.conf` file. Whenever a MUSH starts up or gets rebooted, it's going to look to what's in that file. A few of those choices relate to further configuring the choices you made with `confsource`

The `netrhost.conf` file has some aesthetic options like what if anything it says when the database is being saved or whether or not your MUSH will allow ANSI color, both in general, but also in things like people's names. It has a lot of very esoteric options for tuning the performance and safety of your MUSH. It also is where you define things that connect to your database like your master room and guests. It let's you determine which powers your staff does or does not have, and it's also where you have an option to change the password for #1 should you forget it. There are ways to change some of these options from within the MUSH and even have those changes become new defaults that survive a reboot. One thing you set there that you definitely can't change from within the MUSH, is which port it runs on.

1.3.3 Starting database options

This last big choice is one that you probably are well aware of at least some of the things it allows for, mostly because the database is basically where everyone on a MUSH lives. Most of the choices you ever make about your MUSH will happen in the database and it's something that's basically always going to be changing in more ways than any one person could follow. No matter how vast the database of a MUSH gets, they all started somewhere, and that's the last big choice you have to make.

Depending on how you look at it, there's somewhere between thousands of choices and two choices for a starting database. What I mean is that you have the choice of starting with a brand new database to populate, only a room (#0) and you (#1), even the Master Room is something you'd have to add to it. The other option is to import an existing database, though to choose that option you'd also have to have access to an existing database.

Besides the brand new database that can be made on-demand, RhostMUSH comes with two databases that you can use to get your start. One is called 'Minimal' and the other is called 'Ambrosia' after the lead RhostMUSH developed that made it. Despite the name, they're both fairly minimal, there's no grid in either, but what you get is a lot of pre-installed softcode and security, as well as a `netrhost.conf` file that has been tuned to work well with it. Even if you don't use those databases you can take ideas for the `netrhost.conf` file for tuning your MUSH or even use some RhostMUSH commands to import that softcode into your database.

1.3.4 The Choices We Make

Well, not you're aware of those three big choices, what they are, and where you'll run across them. Later in this Handbook we'll be going through those choices in-depth. In this chapter, what we'll be doing is using the default options for for `confsource` and `netrhost.conf` and a brand new database. Those options and that blank slate are necessary so that you have in front of you a working MUSH with only #0 and #1 and get shown the way of building it into whatever is your dream MUSH, and I know we all have one.

There will be a chance to make all those other choices later, in fact that will be happening very soon. There's a lot of very interesting choices that you can make with RhostMUSH, too many for anyone to ever be able to use them all. If you don't start with mastering the basics, you'll never end up knowing what are the most RhostMUSH options. I can't tell you what they are, it's truly something you have to discover on your own. Remember, you want to build your dream MUSH, not mine.

Note: Unless you are intending to start with a brand new database, you must be aware of it's needs and expectations for the settings of `confsource` and `netrhost.conf` Starter databases tend to distribute with them config files of at least the options they expect set or not set during the process of configuring and compiling the server.

Knowing which database you intend to use is the first choice made of these initial major configuration options, but that last that is made part of the MUSH.

1.4 Compiling RhostMush

1.4.1 Setup directory permissions

In order to both compile and run, all of the RhostMUSH files and directories need to have the proper permissions set. If you obtained the source code directly from GitHub, it is likely that this step is not required, but there is no harm in running it anyway:

```
./dirsetup.sh
```

If you did NOT obtain the source code directly from GitHub, it is possible that even the above script will fail to run with 'permission denied' or similar errors. It is recommended that you obtain the source code from there, but if for whatever reason this is not an option, manually adjust your permissions and then re-run the automated permission script:

```
chmod +rx bin/*.sh src/*.sh game/*.sh game/Startmush game/db_*
./dirsetup.sh
```

1.4.2 Compile the source code

Once the source code has been obtained and the proper file and directory permissions have been set, the RhostMUSH source code is ready to be compiled. This is typically done through an interactive program where you configure the options you want to have available to your installation:

```
make confsource
```

Note: It is recommended that if you are just starting out with RhostMUSH that you compile for the first time using the default compile options which have specifically been tuned to be closest to what the average person would need or expect. Changing these options before you have a grood grasp of what they mean and how RhostMUSH works on a deeper level can potentially cause security issues, reduce compatibility with softcode rom other types of MUSH servers, as well as waste system resources.

Saving your compile options

`make confsource` will remember the most recent options you used to compile the source code for the next time you use `make confsource` It might still be a good idea to more permanently save the options you used to attempt to compile. However, you still might want to have these options saved more permanently, just in case. At the `make confsource` menu there is an option to save your current settings to a file. If you choose to do this, you will find the the save file in the `bin/` directory.

Troubleshooting compile errors

Should this result in an error, a script has been included to correct the most common errors, after which you can once more try to compile:

```
./bin/script_setup.sh
make confsource
```

Once the compile process successfully complete, you should be able to start-up your new RhostMUSH server. If it complains about missing binaries make sure they are linked. The provided script will fix this issue, and is not harmful to run in any situation:

```
make links
```

Recompiling the source code

If you plan to use `make confsource` to recompile your source, you should first issue a `make clean` before re-issuing a `make confsource`

A failure to issue `make clean` prior to re-compiling with `make confsource` can potentially leave stale object files which may cause unforseen issues when running code, including but not limited to random crashes. Generally whenever recompiling it's good to always make clean first.

Note: You may also issue `make source` if the `Makefile` is already defined how you want it to be. Please remember to `make clean` before `make source` whenever you alter the code or import new source code.

1.5 Configuring the game

1.5.1 Persistent configurable game options

Upon compiling a RhostMUSH server, if it doesn't already exist, a `netrhost.conf` is copied into the `game/` directory for your game. It includes useful defaults for most compile-time options that will work well for most games, particularly ones using both the default `confsource` options and related database.

This configuration is derived from `defaults/game/netrhost.conf.default`

While this `netrhost.conf` is very well documented and quite easy to change in some places, but there are also some rather technical options that you it's important to know the full implications of this.

Note: The default `netrhost.conf` starts the game running on the port *4021* of the server. If this is your time creating a MUSH, it is recommended that this setting is the only one that you potentially change, and only if there is a good reason to. Ports below 1024 are priviliged ports and can not be used for this purpose.

1.5.2 Starting the game

Once done, you start up the system with the following command:

```
./Startmush
```

It will prompt you to start a new db if it doesn't find one.

You may also do the commands individually:

```
[csh] netrhost -s netrhost.conf >& netrhost.log &
[sh]  netrhost -s netrhost.conf > netrhost.log 2>&1 &
```

1.5.3 First login to the game

Once started, log in the #1 character (Wizard) with it's appropiate password (no, not 'potrzebie', but 'Nyctasia'). There were private reasons for the password change.

Once in, do a @shutdown to save the database. Then you can run Startup normally. You may make a backup of your database at anytime on-line by utilizing the @dump/flat option. A script comes with this distribution that allows the ability of auto-archiving your database for a configurable number of backups.

1.6 Creating RHostMUSH with a Provided DB

1.6.1 Important before you actually start building

The main parts of making your RhostMUSH, easy pleasy:

1. The stunnel directory contains TLS/SSL connectivity. This has to be linked to another port and will tunnel to the mush port. The README file explains how to set up and configure your TLS/SSL connection.
2. ./patch.sh – This makes sure you have the latest code. If you got this by git clone https://github.com/RhostMUSH/trunk then you can ignore patching. You can use ./patch.sh at any time to update your code. It ignores local.c incase you make your own modules.

3. make confsource. Yup, it's menu driven, nifty eh?
    1. Options you may want to select (other than the defaults):
    2. 5 (%c is selected by default, but choose %x as well for MUX/TM3 compat)
    3. 9 (if you want $commands to require the COMMAND flag)
    4. 16 (if you want a wider WHO listing like older versions of MUX)
    5. 22 (if you're converting a TinyMUSH3 or TinyMUX/MUX2 flatfile)
    6. 24 (if you have issues with -lssl not being found)
    7. B3 (for 64 character attribute names)
    8. B6 (select 8K for Penn/MUX2/TM3 default, up to 32K. 64K is network intensive)
    9. B5 (will be autoselected if you choose 8K or more. Pick this anyway)
    10. B4 (if you have sqlite libraries and wish to use this)
4. 'r' to compile with the settings you selected.
5. Modify your netrhost.conf file as specified. Make sure to align your port and debug_id as shown in the netrhost.conf file.
6. If you wish to port in an old flatfile, please refer to the readme directory on how to port your flatfile in (README.DBLOADING).

1.6.2 Using the prebuilt flatfile

There are pre-loaded flatfile databases you can use at this point. The netrhost.db.flat and corrisponding netrhost.conf file will be located in the minimal-DBs/minimal_db directory.

You may auto-load the minimal db and corresponding netrhost.conf file with the command:

```
./minimal.sh
```

This is ran from within the 'game' directory. Once this is ran, you will need to customize the netrhost.conf file for your purposes. The port and debug_id must be changed at the very least. Keep the debug_id coordinated to the port as described.

To load a prebuilt flatfile

1. Make a backup of your existing netrhost.conf file:

    ```
    cp game/netrhost.conf game/netrhost.conf.backup
    ```

2. Copy the netrhost.conf file into your game directory:

    ```
    cp -f ./minimal-DBs/minimal_db/netrhost.conf ./game/netrhost.conf
    ```

3. At this point you can modify your netrhost.conf file settings in your game directory. Using an editor modify the 'port' and 'debug_id' respectively in your netrhost.conf as state. The 'port' will be the port the mush listens on, the debug_id is for the debug-stack and is your port with a '5' at the end. So if your port is 4444, the debug_id is 44445
4. Load in the flatfile into the mush (You could do this in the Startmush as well) Manually:

```
cd game
```

./db_load data/netrhost.gdbm ../minimal-DBs/minimal_db/netrhost.db.flat data/netrhost.db.new dwF

Start your mush:

```
./Startmush
```

This will load the db that you loaded.

————————OR———-

From Startmush when prompted, hit <RETURN> for searching then select the number of the netrhost.db.flat that is listed as ~/minimal-DBs/minimal_db/netrhost.db.flat:

```
./Startmush
```

1.7 Basic Instructions for starting a new RhostMUSH

1.7.1 Manual configuration of source code

To do manual configuration (skip if the previous step worked for you) And yes, this is a bit of a pain in the bottom, hopefully you will not need this.

You need the following definitions defined to make this work:

1. TINY_U, USE_SIDEEFFECTS, MUX_INCDEC, ATTR_HACK
2. (u()/u2() switched)
3. (sideeffects)
4. (inc()/xinc() switched)
5. (support for _/~ attribs)

You only need to do this if you received the RhostMUSH src. If you received a binary, continue on to the next part.

To compile the code, just type 'make confsource'. It will prompt you with settings on what you need to do. If you just want to quickly hand edit the Makefile, it is in the directory src (full path src/Makefile). Then you may just run 'make source', if you so choose to hand-edit the Makefile.

After the compile process is done, type 'make links'!

1.7.2 Loading a database for your MUSH

You now have a choice of optionally starting at a provided database or starting from scratch.

Option: Only perform these steps if using a provided database

Copy an existing flatfile and corresponding netrhost.conf file Default provied example:

```
cp game/netrhost.conf game/netrhost.conf.backup
cp -f minimal-DBs/minimal_db/netrhost.conf game/netrhost.conf
cd game
./db_load data/netrhost.gdbm ../minimal-DBs/minimal_db/netrhost.db.flat data/netrhost.
↪db.new
```

1.7.3 Option: Things to do once you have connected if you did NOT use a provided database

1. @dig your master room and in your netrhost.conf file define master_room to this dbref (without the #. So like master_room 2)
2. Create an immortal holder charater (@pcreate then @set immortal) Feel free to set up holder characters for all the bittypes which are: GUILDMASTER, ARCHITECT, COUNCILOR, WIZARD, IMMORTAL
3. @chown/preserve the master room and #0 to the immortal holder character.
4. Log into the immortal character
5. @pcreate all your guest characters and set them up properly. My suggestion:

```
@dolist lnum(1,10)={@pcreate Guest##=guest;@set *Guest##=guest;@desc *Guest##=A␣
↪guest player.;@adisconnect *Guest##=home;@lock *Guest##=*Guest##}

@list guest will show your guest characters and if they're set up properly.
```

6. Any master room code you load in from your immholder character (or @chown/preserve to it) The readme directory has softfunctions.minmax that has MUX/Penn compatability functions and comsys. All other softcode (like mail wrappers) can be found on https://github.com/RhostMUSH/trunk in Mushcode.
7. Setup new character, staff, and take tasks that can only be accomplished by #1
8. Set up any other characters you want. Anyone immortal can issue @function, @admin, or anything #1 can do.

1.8 Customtize the textfiles for your game

All connect.txt and customized files can be found in the ~/Server/game/txt directory. There is a README file there that explains their purposes in more detail. You can see more information on all files and how they inter-relate with 'wizhelp file'.

1.9 Three Options for Starting a MUSH

The RhostMUSH Git Repository comes with three options for starting your Mush.

1.9.1 Option 1: Creating a new game with a blank database

Modify your ./game/netrhost.conf file or what settings you want. Don't feel overwhelmed, it's all very well documented.

1.9.2 Options 2: Creating a new game with Ambrosia's default database

Follow minimal-DBs/Amb-MinimalRhost/IMPORTANT_README

The netrhost.conf file you will copy is in minimal-DBs/Amb-MinimalRhost/game Copy this netrhost.conf file into your 'game' directory.

You will want the custom txt files under Amb-MinimalRhost/txt in your game/txt directory and to mkindx all the txt files. You can run ./Startmush -i to index.

When ./Startmush prompts you to load a flatfile, say 'yes' and hit <RETURN> to have it search for flatfiles, then select netrhost.db.flat from under the minimal-DBs/Amb-MinimalRhost directory.

The main steps to make sure you do for ~/Server/minimal-DBs/Amb-MinimalRhost/netrhost.db.flat – Ambrosia's secure and featured minimal db

1. Use the matching netrhost.conf file under the Amb-MinimalRhost/game directory
2. Load in the settings specified in the Amb-MinimalRhost/bin directory.
    1. Copy this file into your ~/Server/bin directory
    2. From 'Server' directory type: make clean
    3. From 'Server' directory type: make confsource and 'l'oad option 0
    4. Specify any -additional- options you want at this point.
    5. Recompile your code
3. Copy the files in Amb-MinimalRhost/game/txt into your ~/Server/game/txt directory
4. from your ~/Server/game txt file run on each of the txt files:

```
../mkindx <txtfile>.txt <txtfile>.indx
```

Note: Where <txtfile> is the name of the file (minus the .txt extension)

1. If running, @reboot your game.

1.9.3 Option 3: Creating a new game with the generic default database

Copy the netrhost.conf from minimal-DBs/minimal_db to your game directory.

When ./Startmush prompts you to load a flatfile, say 'yes' and hit <RETURN> to have it search for flatfiles, then select netrhost.db.flat from under the minimal-DBs/minimal_db directory.

CHAPTER

TWO

WHAT RHOSTMUSH IS ABOUT AND WHAT'S SO GREAT ABOUT RHOSTMUSH

RhostMUSH was founded in 1989, originally by Natasha Davis (Nyctasia) and as a branch from the original release of TinyMUD code. It was her desire to make a game that was flexible, with multiple levels of progression and highly customizeable. She lost time and interest and passed the game to Steve Shivers (Seawolf), Mike McDermott (Thorin), and Jace Hoppel (Ashen-Shugar)

Through their work, the stability improved, we fixed it to be multi-platform and as bug free as we could possibly make it. We introduced several methods both in game and in source that allowed consistent memory bounds checking and various alerts for any mischievous naughtyness in-game or possibilities of any hacks, patches, or alterations in the code causing leaks or issues.

While not perfect, it has allowed us to have an absolutely outstanding turn around for any bugs sent our way, which anyone who uses RhostMUSH will attest to.

Over the years, others have joined the RhostMUSH team, including Ambrosia (who is the current dev lead), Lensman, Kage (who kindly provided the UTF8/unicode port), Jeff/Loki, Rook, Noltar, and Odin.

We also have had hundreds of people who have offered (and provided) help, patches, suggestions, bug fixes, and alternations all on their own and every single one will have had their name mentioned in the RHOST.CHANGES file in the readme directory. It's far too large to have in the online help files.

RhostMUSH today provides an amazing tool that allows nearly entire customization in-game of every single feature available in Rhost without having the requirement to modify the hardcode. This includes but is not limited to:

2.1 Recycle bin

Yup, you guessed it. RhostMUSH has a windows like recycle bin. This means the objects you @nuke and @destroy become 'destroyed' but not recycled until they are @purged. If you use the Myrddin CRON in the Mushcode directory, it by default sets up a job to purge anything over 30 days old, which should be more than sufficient for any needs. The goodness of this? You can recover nuked things from any period of time, as long as they were not @purged first.

Commands: @purge, @nuke, @destroy, @recover, @reclist

2.2 @snapshot

Live image snapshots to unload or load to and from disk. As many snapshots as you want, as often as you want. It essentially does a flatfile dump of a dbref#. Great for backups or cross-Rhost portability.

Command: @snapshot

2.3 Wizard and Immortals by default

- are spoofable. Meaning all their @pemits by default will not trigger NOSPOOF. If you do not wish this, set the SPOOF flag this applies to anyone below their level.
- override all locks. There's two flags to disable this. NO_OVERRIDE to stop overriding all locks (including attribs) and NO_USELOCK to just stop overriding uselocks. This applies to anything their level and lower.
- optionally cloak from all non-immortals/God player. This can be highly abused if not careful and there is a @depower to disable cloaking and/or dark that will disable this.
- immortals can optionally supercloak from even wizards. this can not be disabled, and you must consider that immortals should be treated as the God player (#1) since they are effectively #1 in nearly every way.

2.4 Titles and Captions to a player's name

@caption and @titlecaption

2.5 Have an alternate name with locks for NPC obfuscation

@altname @lock/altname

2.6 Have multiple player aliases

As well as a method to reserve player names per player w/o revealing who has what name.

@protect

2.7 Actively control how dark works both game-wide and individually

@depower dark

@admin allow_whodark, sweep_dark, command_dark, lcon_checks_dark, secure_dark, see_owned_dark, idle_wiz_dark, player_dark

@toggle snuffdark

@flagdef to redefine who and what can set the DARK flag

2.8 Make config file changes in-game without having to reboot or have shell access

@admin admin_object

2.9 Execute any binary or script as a localized function

EXECSCRIPT (power), SIDEFX (flag)

2.10 Customized percent substitutions (like %n, %#, etc)

@admin sub_include, @hook

2.11 Redefine percent substitutions (like %n, %#, etc)

@admin sub_override, @hook

2.12 Localize command and function overrides in a sandbox

@icmd, @lfunction, subeval(), sandbox()

2.13 Multiple Zones

Have multiple zones which can optionally belong to multiple targets (multiple zones per target allowable!)

@zone, zones, lzone(), zonecmd()

2.14 Optionally control, enable, or disable sideeffects

@admin sideeffects, SIDEFX (flag)

2.15 Have 31 cross-interactive realities for locations

This works as a truly independant and self-contained environment. A room can have 31 'layers', each 'layer' is a reality in the same physical space. These layers can work independently or allow interaction with other layers for vast customization. This affects all methods within the game including all matching, looking, $commands, listens, movement, interaction, pretty much every single aspect of mushing.

REALITY LEVELS

2.16 Override any command with softcode

@admin access (check ignore)

Master room $commands to then override the hardcode

2.17 The abilility to raise or lower permissions on the various

staff bitlevels for each player.

@power, @depower, TOGGLES, FLAGS

2.18 Customize new commands on the connect screen

@admin file_object2

2.19 Softcode any txt file (like connect.txt)

and have it evaluate in-game. It evaluates as the object it is on.

@admin file_object

2.20 Advanced tracing methods for debugging your code including labels!

Commands: @label

Functions: parenmatch(), trace()

Toggles: CPUTIME

Flags: TRACE

Attributes: TRACE_GREP, TRACE, TRACE_COLOR, TRACE_COLOR_<attr>

Substitutions: %_

2.21 Built in pretty-printing of attributes with the parenmatch() function

Example Code Output:

```
@emit [add(1,sub(2,div(3,4)),5)];@emit [extract(get(me/foo),3,1)

Example Pretty Print:
@emit [
   add(
      1,sub(
```

(continues on next page)

(continued from previous page)

```
          2,div(
             3,4
          )
       ),5
   )
];@emit [
   extract(
      get(
         me/foo
      ),3,1
   )
]
```

2.22 Plenty more not mentioned!

The flexibility to customize RhostMUSH is what is most daunting. Don't fret, you don't need to do it to run RhostMUSH successfully. In fact, the default configuration is mostly compatible with MUSH and will work correctly out of the box for most needs. For those wishing to play, of course the sky is the limit of what you want to do.

2.23 Advanced features of RhostMUSH

2.23.1 Debugging/Tracing

- Debugging in Rhost allows for advanced features like expressing where and when to do debugging via a trace() function, with toggled labels, and the ability to grep content from trace output. There also exists features to color-match parenthesis, braces, and brackets in-game as well as pretty print the output of commands and functions.
  - help trace
  - help %_
  - help trace()
  - help parenmatch()
  - help parenstr()

2.23.2 Zoning

- Zoning in Rhost allows the same functionality of Penn and MUX, though the syntax is different. It also allows the ability to belong to multiple zones at the same time and take advantage of mulitple zones at once. This allows for increased levels of complexity.
  - help zones
  - help @zone
  - help zonecmd()
  - help lzone()
  - help @Lock type twink

  - help @lock type zone

2.23.3 Reality Levels

- Reality levels allows for the ability to have a sandboxed 'existance' in each physical location across the entirity of the mush. Each reality is its own sandbox and can either stand alone or work dependently with other realities. A person can belong to multiple realities at the same time, and realities is geared to a method for send and receive. Each 'method' requires to be in the given reality to affect it.
  - help reality levels
  - wizhelp chkreality
  - wizhelp reaity level
  - help @Lock type user

2.23.4 Function and Command Overriding

- Functions and commands can both be overridden with softcode. To override a hardcoded command you first set the command ignore. There are various levels of ignoring so that you could have it ignored from mortals but have it executed fine for non-mortals. This allows you to use the actual physical command within a softcode override. You may also use @Hook for altering how a command works. Functions are overridden by setting the function in question ignored, then writing a softcode alternative that is then executed and fetched appropriately.

  Commands:
  - wizhelp @admin
  - wizhelp access
  - wizhelp permissions
  - wizhelp @Hook
  - wizhelp hook setup

  Functions:
  - wizhelp @admin
  - wizhelp function_access
  - wizhelp @function
  - help @lfunction
  - wizhelp bypass()

2.23.5 The Recycle Bin

- Rhost has a recycle bin that works a bit like a windows recycle bin. Whenever you destroy something within the mush, it is stacked onto the recycle bin and marked unavailable within the mush. This marks the dbref# as garbage in any sense of the word. However, the object is not able to be reused until purged. Once purged, it is put onto a free list that can then be reassigned to a new object.
  - wizhelp @nuke
  - wizhelp @destroy
  - wizhelp @toad

  - wizhelp @turtle
  - wizhelp @purge
  - wizhelp @recover
  - wizhelp @reclist

2.23.6 Percent Substitution Adding/Overriding

- Rhost allows the ability to both override percent substitutions as well as creating new ones. This is done with @Hook and admin params and issues softcode overriding. Due to how it is evaluated there is no risk of recursion.
  - wizhelp @hook
  - wizhelp hook_cmd
  - wizhelp sub_include
  - wizhelp sub_override

2.23.7 Hooking

- Hooking allows you to have advanced methods to manipulate commands including adding customized switches to them via softcode.
  - wizhelp @hook
  - wizhelp hook_cmd
  - wizhelp hook_obj
  - wizhelp hook setup

2.23.8 Command based uselocks

- This allows you to have unique uselocks per $command. This is done through the use of the USELOCK attribute flag, then you set up a matching attribute name with a prefix of a ~ to specify how the lock is to be evaluated. This works in the same manner as an evaluation lock. To be able to use the USELOCK attribute flag you must be empowered to do so with the 'ATRUSE' @toggle. You may also use the secure_atruselock config parameter to globally enable this and not require the toggle to be set.
  - wizhelp atruse toggle
  - help attribute uselocks

2.23.9 Differentating between command and listen locks

- We distinguish between commands and listens with uselocks by passing an optional argument to all locks that are uselocks. This optional argument is 0 for a default lock, 1 for a command lock and 2 for a listen lock.
  - help @lock type uselock

2.23.10 Wizard auto-overiding and how to disable it

- By default wizards override all locks, including attribute locks, can see all dark exits, and bypass pagelocks. This can be troublesome, and even abusive, so there's ways to disable this.
  - wizhelp @depower (for those abusing it)
  - wizhelp no_override (disable overiding locks)
  - wizhelp no_uselock (disable just uselock overriding)
  - wizhelp pagelock toggle (disable pagelock overriding)

2.23.11 Advanced FLAG and TOGGLE control

- Flags and toggles can be controlled to have multiple permissions and enable/disable targets of how the flags are allowed to be set. This is done through commands in-game or you can use conf file options to do so.
  - wizhelp @flagdef
  - wizhelp @toggledef
  - @admin @flagdef alternatives
    - wizhelp flag_access_set
    - wizhelp flag_access_unset
    - wizhelp flag_access_see
    - wizhelp flag_access_type
  - @admin @toggledef alternatives
    - wizhelp toggle_access_set
    - wizhelp toggle_access_unset
    - wizhelp toggle_access_see
    - wizhelp toggle_access_type

2.23.12 Advanced site control

- We allow advanced site control by not only blocking various sites but we can specify how many times a player can be connected at the same time as well as how many times sites are able to connect at the same time. This is done through normal site manipulation.
  - wizhelp forbid_host
  - wizhelp register_host
  - wizhelp noguest_host
  - wizhelp @list (site option)

2.23.13 Auto-Registration

- Autoregistration is the method that a player can auto-register by providing their email on the connect screen. It will email them a password and an optional document that the administrator provides. This is well described in the wizhelp.
  - wizhelp autoregistration

2.23.14 Which bit level is best?

- This is something that should be discussed by you and the staff of your game. As a good rule of thumb, only provide the bitlevel that is required to do the job. Too much power is always risky. In essence, each tier of bit can do everything the previous bitlevel can do, and then additional stuff on top of it. The highest bitlevel is #1 itself, being bitlevel 7. Then immortal, which should be considered the #1 character in most cases and is bitlevel 6. Then the royalty character, which is equal to wizard on penn, mux, or other codebases. For most things, this is the bitlevel you want to assign players. The exception will likely be game owners or people who control the master room code.

  There's a bunch of readme files and online wizhelp that goes into detail of the various bitlevels and what each can do.
  - wizhelp control

2.23.15 What are the limits for size/growth for RhostMUSH?

- While using QDBM, there's really no set limits for most things. The limits that we have are as followed:

  LBUF - 64K. It is recommended to only use 32K as there is some issues with networking with 64K lbufs.
    - Compile time option with the menu configurator

  SBUF - 64 characters (if configured – it's suggested you do).
    - Compile time option with the menu configurator

  MBUF - 200 characters. Not able to be changed.

  MAX CONNECTIONS - Limited by the total number of open sockets and descriptors on the account and server running on. There are various tools to limit connection DoS attemps and other such nastiness. This is well documented in the netrhost.conf file.
    - wizhelp max_players
    - ~/game/netrhost.conf

  MEMORY - no limit. Generally runs between 8-50M depending on the size of the mush and the LBUF size specified.

  CPU - no limit, but has built in cpu abort in code. The netrhost.conf file documents this well for customizing. the default values are usually good enough.
    - wizhelp max_cpu_cycles
    - wizhelp cpuintervalchk
    - wizhelp cputimechk
    - wizhelp cpu_secure_lvl

- wizhelp heavy_cpu_max
- ~/game/netrhost.conf

DISK - no limit. Generally will be 75-200M depending on size, number of backups and if you leave your compiled object files in.

DB Size - (20000 default) There is no limit on the number of objects the db can have. By default it's soft limited to 20K objects, which can be changed by a netrhost.conf file change. We have had this up past 1.5 million objects, and other than a second or two of lag for complex searches we had no real problem.

- wizhelp maximum_size
- help @quota
- wizhelp @quota
- wizhelp @limit

Attribute Size - 10K as a hard limit. 750 as a soft limit. You can increase this but it can't exceed 10000 attributes. This is to avoid DoS style attacks.

- wizhelp vlimit
- wizhelp @limit

2.23.16 Sqlite and MySQL/Maria setup and why use it?

- Both of these can be configured separately or conjointly to run in parallel. This can be done through the RhostMUSH configuration utility. You generally want to use SQL for external data storage or accessing a central repository of data to share between multiple projects. Like, for example between a wiki, a forum, and the mush.

2.23.17 Executing outside scripts and binaries within RhostMUSH

- Rhost has execscript() which allows executing outside binaries or scripts as a native function. All effort has been done to avoid any type of DoS based issue or hang by doing this, however the guidelines presented should be followed before doing so.
  - wizhelp execscript
  - wizhelp power execscript
  - help sidefx
  - wizhelp writing scripts

2.23.18 Pulling external data into RhostMUSH

- You are capable of pulling external data in to RhostMUSH using several methods. These are by using:
  - SQL (mysql or sqlite)
  - execscript
  - cron (wizhelp signal)

2.23.19 Integrating a unix cron right into RhostMUSH

- The unix cron can be used to integrate with Rhost fairly easilly by use of signals. By using SIGUSR1 you can specify Rhost to execute code in-game, which part of that could be to pick up a pre-designed list of commands that the unix cron has set up.
  - wizhelp signal
  - wizhelp signal_cron

2.23.20 Signal handling, how it works, and when and why use it

- Signals are used to do different things for the mush. By default, the following signals are recognized by the mush and will do the following as defaults.
  - SIGUSR1 - will do a reboot of the mush. This is also customizable so that you can have it execute code in-mush if you want.
  - SIGUSR2 - will do a clean shutdown of the mush.
  - SIGTERM - will immediately scram the mush as cleanly and fast as possible. It will avoid dumping anything to the database to speed up scramming, but write a TERM flat file to be loaded in if corruption.

2.23.21 Setting up global parents, global @parents, global attribute formatting, and other global setups

- Global parents are useful when you want to have a global 'parent' without actually having a defined '@parent'. It always will be the highest tier in a lookup. The way lookups will go will be:

```
self->@parent(s)->@zone(s)->GlobalParent
```

  The type of the parent does not have to match the target.

  These global parents can be defined either by using a global generic parent or by using the type. If a type is specified it overrides the generic. The following parameters are used:

  - global_parent_obj - The generic global parent (if defined)
  - global_parent_room - The room global parent
  - global_parent_exit - The exit global parent
  - global_parent_thing - The thing global parent
  - global_parent_player - The player global parent

- Global @parents are different than global parents in that any new item of similar type that is created is automatically assigned this physical @parent. It's obviously more limiting since it sets the literal physical parent defined.

  The type of the parent does not have to match the target.

  The following parameters are used:

  - room_parent - The target that new rooms are @parented
  - exit_parent - The target that new exits are @parented
  - thing_parent - The target that new things are @parented
  - player_parent - The target that new players are @parented

- Global attribute formatting is a method define a wrapper, of sorts, where attributes like @desc, @odesc, @succ, and anything similar can be processed through this. All attributes will be either &FORMAT<attribute> or &<attribute>FORMAT based on the current configuration. Example: &FORMATDESC or &DESCFORMAT localy, or use the following global objects for global formatting. Local formatting has priority.

  The type of the parent does not have to match the target.

  - room_attr_default - Target for room formatting
  - exit_attr_default - Target for exit formatting
  - thing_attr_default - Target for thing formatting
  - player_attr_default - Target for player formatting

2.23.22 RhostMUSH limitations and how to get around them

While Rhost is insanely configurable and quite powerful, there are limitations that exist within it.

- Function invocations. Sometimes you will hit a ceiling on evaluation. You may want to tweak values to allow more functions or commands to execute. The following controls that:
  - function_invocation_limit [25000 default] - specifies the total functions you can execute per command.
  - function_recursion_limit [50] - specifies the total times a function can call itself over and over. Rarely should this be increased and doing so can effect your stack depth.
- Command queue limits. Sometimes you want more to be queued up for players or wizards.
  - player_queue_limit - Max number of entries a player can queue
  - wizard_queue_limit - Max number of entries a wizard can queue
- @limit is a wonderful way to lock down limitations per player or global. Lots of power is available here.
  - @limit
  - vattr_limit_checkwiz - Enable @limit checks for wizards
  - wizmax_vattr_limit - Set wizard global VATTR limits
  - wizmax_dest_limit - Set wizard global @destroy limits
  - max_vattr_limit - Set player global VATTR limits
  - max_dest_limit - Set player global @destroy limits
- Lots of trace output can be cut off. You can modify this with:
  - trace_output_limit - Set lines of trace output shown
- To define how many commands a minute a player set SPAMMONITOR can use
  - spam_limit – default 120
- If you examine things and see 'Output cut off' messages, you may want to increase your output limit, funny enough, the name of this is similar
  - output_limit - You should set this no less than 4 times the current size of your LBUF.
- Attributes names can not exceed 64 characters. Sorry, it's a hard limit
- Sometime you may find a single may not work for an escape. You can in most cases use a % instead or double escape the to make it work. Also look at lit() as a solution.

2.23.23 Advanced guest setup

- After you set up your guests, you can set unique names to each guest if you so want after defining the dbref#'s your guests use. This is done by defining them in the guest_namelist parameter. You can also increase guests (or decrease them) between 0-31 guests.
  - wizhelp guest_namelist
  - wizhelp num_guests

2.23.24 Attribute permission masking and the joys of the power behind it

- Attribute contentlocks can be set up so you can lock the actual content that you can set (or even unset!) into an attribute. The beauty of this is that you can specify case sensitive information, lock different ways contents in attributes are set based on who is setting it, or even on where it's being set. The sky's the limit.
  - global_attrdefault - Target for defining content locks

2.23.25 The amazing @cluster and what it can do for you

- Clusters is the way to virtually assign multiple objects into a single physical object. It essentially chains together two or more objects to share attributes between them, so that any attribute set on any object in that cluster can be set or fetched as if it was a singular entity. This allows some amazing ability to distribute attribute content or even have a farm of a massive amount of attributes without paying a hefty penalty on object bloat.
  - help cluster – Gives a fantastic overview of how clusters work.

2.23.26 What we plan for the future

- Things to look forward to the future with RhostMUSH.
  - Full Unicode/UTF8 in Rhost 4.0
  - A fully featured tag system in Rhost 4.0
  - Built in Python API handler in Rhost 4.0
  - Hopefully a built in Ruby and Perl API in Rhost 4.0/4.1
  - Cross-Mush execution between mushes in Rhost 4.1
  - More as we think about it :)

2.23.27 Additional features not covered otherwise

- +/- 5.4 million years can be utilized with the built in time functions which includes timefmt(), secs(), convtime(), convsecs(), and moon(). Party on!
- Changing permission levels in the middle of execution for evaluation.
  - see help on the streval and ueval function'
- Full features in-game customization of near every aspect of the game.

2.24 Random notes and things to know about RhostMUSH

Here are some things to know about RhostMUSH and what you may or may not want to do. Things here are not covered in other documents:

Admin toggles to configure the WHO, various things you're used to, etc is in the 'netrhost.conf' file. descriptions Notes in the autoconf.h file is in the README.AUTOCONF file.

2.24.1 Note on bits, their levels, and things they do

IMMORTAL - They can do anything. Treat this as #1 and only give to people you trust. Period. You don't have to use this bit if you do not want to and just assume #1.

ROYALTY - Unlike PENN/MUX, this is *not* a sub-wizard, this is a FULL wizard. Plus, they can do a bit more.

COUNCILOR - Like royalty on PENN/MUX but they can modify.

ARCHITECT - Can't do as much as councilor, but lot more than BUILDER.

GUILDMASTER - Very limited. Sees dbrief#'s, can ex things their level and lower and @quota players.

2.24.2 You need to @pcreate your guest characters and set them GUEST

It doesn't create them on the fly but we considered this better. You have 31 total you can have. It defaults to 10 in the netrhost.conf file. You can rename the guests anything you want, but before you do so, you must add the dbref#'s to the param guest_namelist

2.24.3 @powers are INHERITED

Therefore, you need power_objects enabled (@admin) to make this work properly for non-plauyers. A power is taken before a bit level ONLY if higher than that bit. Yes, powers are multi-level.

2.24.4 @depowers are automatically checked first before anything else

This is also meaningless on objects.

2.24.5 Zones are unique

You can have things in multiple zones.

2.24.6 The db auto-repares itself when it can

It does this by purging anything it can't identify. Dataloss is better than unrecoverable data. Yes, any such 'repairing' is logged so you know if something is up.

2.24.7 You can get your connect.txt to parse ansi

See ansi_txtfiles in wizhelp.

You can also override it with softcode if you so wanted.

See file_object in wizhelp for more information on this.

2.24.8 Re-compiled binaries do not require an @shutdown

When re-compiling the binaries, all you have to remember is when done, issue

@reboot on the game and @readcache.

You do not need to @shutdown.

2.24.9 Softcode emulations of functions from other servers are included

Load the file softfunctions into the mush once it's set up. This are @functions that will alias the functions that PENN, MUX, and TinyMUSH have that is either named differently or we don't have for one reason or another.

CHAPTER

THREE

COMPARISON OF MODERN MUSH SERVERS

3.1 Differences to expect to the end user between PennMUSH, MUX, and RhostMUSH

The largest end user experience will mostly resolve around some look and feel. The general look and feel of how to set attributes, work with commands and functions, and getting around a grid will be identical between the mush flavors. @nuke and @destroy will work somewhat differently between the three codebases and some effort should be looked at on how different it behaves.

3.1.1 Differences with PennMUSH

- The way the comsystem (hardcode) works with the latest PennMUSH has some compatibility now with MUX's comsystem.
- The @mail system is different to MUX and Rhost.
- Debugging code uses DEBUG and has an indented format.
- The parser for code doesn't require nested escaping like other codebases, but has issues with the pre-parser and nesting.
- All standard movement, attribute setting, other should be similar
- Penn allows empty attributes.
- Penn supports attribute trees.
- Penn has the standard set of bitlevel flags with on/off @powers

3.1.2 Differences with MUX

- The comsystem (hardcode) is unique to MUX/TinyMUSH3 and is not compatible with Penn.
- The @mail system is unique to MUX/TinyMUSH3 and is not compatible with Penn.
- Debugging uses TRACE and is the old standard for debugging.
- The parser requires extra escaping for nested escaping but handles pre-parsing and nesting fine.
- MUX does not allow empty attributes.
- MUX does not support attribute trees.
- MUX has the standard set of bitlevel flags with on/off @powers

3.1.3 Differences with RhostMUSH

- The comsystem (softcode) is in the Mushcode directory and is compatible to both PennMUSH and MUX.
- The mail system is unique to Rhost but there exists mail wrappers that allow MUX and Penn compatibility.
- Debugging uses TRACE and is the old standard for debugging. It allows advanced features like labeling and grepping for content.
- The parser requires extra escaping for nested escaping but handles pre-parsing and nesting fine.
- Rhost does not allow empty attributes.
- Rhost marginally supports attribute trees.
- Rhost has a multi-level set of bitlevel flags with multi-tier @powers, @depowers, and varying other tools for permissions.

3.2 Comparison of features between RhostMUSH, PennMUSH, and MUX2

There are differences in the initial set up from Rhost to PennMUSH (and MUX2).

A lot of people use PennMUSH and if you are one of them, this may help you. Some people use MUX2, hopefully this will help you as well.

NOTE: If you plan to use sideeffects, you must set the SIDEFX flag to allow the target to use it

NOTE: Rhost does support UNICODE/UTF8, but it's currently not part of the main branch as of this writing. Please talk to Kage on the dev site for more information on when it'll be released.

3.3 Organization of help and wizhelp

First, on PennMUSH, help is separated into 'help' and 'wizhelp'. This can be confusing. If you want to 'combine' them, you may use the following snippit:

$ahelp*:@pemit %#=[switch([!!words(%0)][match(/search,%0*)],0?,[setq(0,help)][setq(1,0)],11,[setq(0,rest(%0))][setq(1,2)],setq(0,tri
entry*),%qa[setq(2,%r)])][ifelse(!strmatch(%qb,No entry*),%q2%qb[setq(2,%r)])][ifelse(!$r(2),No topics found for '%q0'.)]

This will display help from help and wizhelp for any matching topic. It should make life easier for you. This also honors the /search switch.

3.4 Mail, comsystem, guests, master rooms, etc..

The comsystem is softcode and included in the readme directory. (comsys). Just load it in. The mail wrappers can be gotten from:

The Mushcode directory in the Rhost Distribution.

Please see the other readme files on setting other things up like guests, master rooms, and so forth.

3.5 Comparisons of features

Comparisons were done based on the following: RhostMUSH 3.9.5p2 MUX 2.12.0.2 PennMUSH 1.8.6p0

3.6 @commands with equivalents

The following are used for compatibility to keep in mind:

@commands Pennmush————————-RhostMUSH————————————-MUX2 @@ @@ @@ @ALLHALT @halt/all @HALT/ALL @ALLQUOTA @quota/all @QUOTA/ALL @ALIAS @alias/@protect @ALIAS N/A @apply_marked @APPLY_MARKED @ASSERT @assert @ASSERT @ATRCHOWN @chown @CHOWN @ATRLOCK @lock @LOCK @ATTRIBUTE @attribute @ATTRIBUTE N/A (see backup_flat.sh) @BACKUP @BOOT @boot @BOOT @BREAK @break @BREAK N/A N/A @CCHARGE N/A N/A @CCHOWN @CEMIT (softcode) @CEMIT @CHANNEL N/A N/A @CHAT N/A N/A @CHOWNALL @chownall @CHOWNALL @CHZONE @zone @CHZONE @CHZONEALL @zone N/A @CLOCK N/A N/A @CLONE @clone @CLONE N/A N/A @CCREATE N/A N/A @CDESTROY N/A N/A @CWHO @COMMAND N/A N/A @CONFIG @admin @ADMIN N/A N/A @COFLAGS N/A N/A @CPFLAGS N/A N/A @CSET @CPATTR @cpattr @CPATTR @CREATE @create @CREATE N/A @cut @CUT @DBCK @dbck @DBCK @DECOMPILE @decompile @DECOMPILE @DESTROY @destroy @DESTROY @DIG @dig @DIG @DISABLE @disable @DISABLE @DOING @doing @DOING @DOLIST @dolist @DOLIST @DRAIN @drain @DRAIN @DUMP @dump @DUMP @EDIT @edit @EDIT @ELOCK @lock/enter @LOCK/ENTER @EMIT @emit @EMIT N/A N/A @EMAIL @ENABLE @enable @ENABLE @ENTRANCES @entrances @ENTRANCES @EUNLOCK @unlock/enter @UNLOCK/ENTER N/A @eval @EVAL N/A @femit @FEMIT N/A @fpose @FPOSE N/A @fsay @FSAY @FIND @find @FIND @FIRSTEXIT N/A N/A @FLAG @flag @FLAG @FORCE @force @FORCE N/A folder @FOLDER @FUNCTION @function/@lfunction @FUNCTION @GREP @grep N/A @HALT @halt @HALT @HIDE @hide N/A @HOOK @hook @HOOK @INCLUDE @include N/A N/A @skip/ifelse @IF @KICK @kick @KICK N/A @last @LAST @LEMIT @lemit @LEMIT @LINK @link @LINK @LIST @list @LIST N/A @list_file @LIST_FILE @LISTMOTD @listmotd @LISTMOTD @LOCK @lock @LOCK @LOG @log @LOG @LOGWIPE N/A N/A @LSET @set @SET N/A @mark @MARK N/A @mark_all @MARK_ALL @MAIL mail @MAIL @MALIAS wmail/alias @MALIAS @MAPSQL N/A N/A @MESSAGE @pemit/@remit + parsestr() N/A @MONIKER @extansi @MONIKER @MOTD @motd @MOTD @MVATTR @mvattr @MVATTR @NAME @name @NAME N/A @emit/noeval @NEMIT N/A @pemit/noeval @NPEMIT @NEWPASSWORD @newpassword @NEWPASSWORD @NOTIFY @notify @NOTIFY @NSCEMIT N/A N/A @NSEMIT @emit @emit @NSLEMIT @lemit @LEMIT @NSOEMIT @oemit @OEMIT @NSPEMIT @pemit @PEMIT @NSPROMPT N/A N/A @NSREMIT @remit @REMIT @NSZEMIT @zemit N/A @NUKE @destroy/@nuke @DESTROY/@NUKE @OEMIT @oemit @OEMIT @OPEN @open @OPEN @PARENT @parent @PARENT @PASSWORD @password @PASSWORD @PCREATE @pcreate @PCREATE @PEMIT @pemit @PEMIT @POLL @doing/header @POLL @POOR @poor @POOR @POWER @power @POWER @PROMPT N/A (@program?) N/A (@program?) N/A @program @PROGRAM @PS @ps @PS @PURGE @timewarp/dump 1 @TIMEWARP/DUMP 1 N/A @quitprogram @QUITPROGRAM @QUOTA @quota @QUOTA N/A N/A @QUERY @READCACHE @readcache @READCACHE @RECYCLE @purge N/A N/A N/A @REFERENCE N/A @robot @ROBOT @REJECTMOTD @rejectmotd @REJECTMOTD @REMIT @remit @REMIT @RESTART @reboot @RESTART @RETRY N/A N/A @RWALL @wall/wiz @WALL/WIZ @SCAN (see softcode) N/A @SEARCH @search @SEARCH @SELECT @switch/first @SWITCH/FIRST @SET @set @SET @SHUTDOWN @shutdown @SHUTDOWN @SITELOCK @admin forbid_host/forbid_site @admin forbid_site @SLAVE N/A @STARTSLAVE @SOCKSET N/A N/A @SQL (only if MySQL enabled) @QUERY @SQUOTA @quota N/A @STATS @stats @STATS @SWEEP @sweep @SWEEP @SWITCH @switch @SWITCH N/A @timewarp @TIMEWARP @TELEPORT @teleport @TELEPORT N/A @timecheck @TIMECHECK N/A @toad @TOAD @TRIGGER @trigger @TRIGGER @ULOCK @lock/use @LOCK/USE @UNDESTROY @recover N/A @UNLINK @unlink @UNLINK @UNLOCK @unlock @UNLOCK @UNRECYCLE @recover N/A

@UPTIME @uptime @UPTIME @UUNLOCK @unlock/use @UNLOCK/USE @VERB @verb @VERB @VERSION @version VERSION @WAIT @wait @WAIT @WALL @wall @WALL @WARNINGS N/A N/A @WCHECK N/A N/A @WEBPASSWD N/A N/A @WHEREIS @whereis N/A @WIPE @wipe @WIPE @WIZMOTD @wizmotd @WIZMOTD @WIZWALL @wall/wiz @WALL/WIZ @ZEMIT @zemit N/A N/A (softcode) ALLCOM N/A (softcode) COMLIST N/A (softcode) DELCOM N/A (softcode) ADDCOM N/A (softcode) COMTITLE ANEWS @dynhelp N/A ATTRIB_SET (@hook on S) (@hook on S) BRIEF ex/brief EX/BRIEF BUY N/A N/A N/A N/A CLEARCOM DESERT (see follow softcode) N/A DISMISS (see follow softcode) N/A DOING doing DOING DROP drop DROP EMPTY @tel/list lcon(target)=me @tel/list lcon(target)=me ENTER enter ENTER EXAMINE examine EXAMINE FOLLOW (see follow softcode) N/A GET get GET GIVE give GIVE GOTO goto GOTO HELP help/wizhelp HELP/WIZHELP HOME home HOME HUH_COMMAND @admin global_error_obj @admin global_error_obj INFO INFO INFO INVENTORY inventory INVENTORY KILL kill KILL LEAVE leave LEAVE LOGOUT logout LOGOUT LOOK look LOOK NEWS news NEWS N/A outputprefix OUTPUTPREFIX N/A outputsuffix OUTPUTSUFFIX PAGE page/lpage/rpage/mrpage PAGE POSE pose POSE N/A N/A PUEBLOCLIENT QUIT quit QUIT N/A N/A REPORT SAY say SAY SCORE score SCORE SEMIPOSE pose/nospace POSE/NOSPACE SESSION session SESSION SLAY slay SLAY TEACH train TRAIN THINK think THINK UNFOLLOW (see follow softcode) N/A UNIMPLEMENTED_COMMAND @admin global_error_obj @admin global_error_obj USE use USE WARN_ON_MISSING N/A N/A WHISPER whisper WHISPER WHO who WHO WITH N/A N/A

3.7 @commands unique to RhostMUSH

Commands that exist in Rhost that have no PennMUSH equivelant: @aflags @apply_marked @areg @blacklist @cluster @conncheck @convert @cut @dbclean @depower @dynhelp @eval @femit @fixdb @fpose @freeze @fsay @icmd @last @lfunction @limit @logrotate @mark @mark_all @money @pipe @program @progreset @protect @quitprogram @reclist @recover @register @remote @robot @rxlevel @skip @snapshot @snoop @thaw @timewarp @toad @toggle @toggledef @tor @turtle @txlevel @whereall grab join listen mrpage newsdb rpage smell taste touch wielded worn +players

3.8 @lock equivalents

@locks PennMUSH————RhostMUSH————————————MUX2 BASIC BASIC/DEFAULT DEFAULT ENTER ENTER ENTER TELEPORT TPORT TPORT USE USE USE PAGE PAGE PAGE ZONE ZONEWIZLOCK/ZONETOLOCK/TWINKLOCK N/A PARENT PARENT PARENT LINK LINK LINK OPEN OPEN OPEN MAIL mail/lock MAIL USER USER USER USER:<dynamicname> lockencode()/lockdecode()/lockcheck() N/A SPEECH SPEECH SPEECH LISTEN USE (see listen argument) N/A COMMAND USE (commands are default) N/A LEAVE LEAVE LEAVE DROP DROP DROP DROPIN DROPTO N/A GIVE GIVE GIVE FROM GIVETO N/A PAY N/A N/A RECEIVE RECEIVE RECEIVE FOLLOW (See softcoded follow code) N/A EXAMINE See NO_MODIFY/NO_EXAMINE/TWINKLOCK N/A CHZONE ZONETOLOCK/ZONEWIZLOCK/TWINKLOCK N/A FORWARD N/A N/A FILTER N/A N/A INFILTER N/A N/A CONTROL CONTROL N/A DROPTO DROPTO N/A DESTROY See: @recover/@purge/INDESTRUCTIBLE/SAFE N/A INTERACT N/A (See: Reality Levels) N/A (See: Reality Levels) TAKE GETFROM GETFROM MAILFORWARD mail/lock, mail/autofor N/A N/A TELOUT TELOUT N/A DARK VISIBLE

3.9 @locks that only exist in RhostMUSH

@locks that exist in Rhost that have no PennMUSH equivelant: TELOUTLOCK TWINKLOCK DARKLOCK ALTNAME CHOWN

3.10 Flag and toggle equivalents

FLAGS Pennmush————————-RhostMUSH————————————MUX2 ABODE ABODE ABODE N/A N/A ASCII ANSI ANSI ANSI AUDIBLE AUDIBLE AUDIBLE (Not Needed) (Not Needed) BLEED N/A AUDITORIUM AUDITORIUM N/A BLIND BLIND N/A COMMANDS COMMANDS CHAN_USEFIRSTMATCH N/A N/A CHOWN_OK CHOWN_OK CHOWN_OK CLOUDY N/A N/A COLOR ANSICOLOR N/A CONNECTED CONNECTED CONNECTED DARK DARK DARK DEBUG TRACE TRACE DESTROY_OK DESTROY_OK DESTROY_OK ENTER_OK ENTER_OK ENTER_OK FIXED NO_TEL FIXED FLOATING FLOATING FLOATING GAGGED FUBAR GAGGED GOING GOING GOING HALT HALT HALT HAVEN HAVEN HAVEN (see @flag) (marker0-marker9) HEAD HEAR_CONNECT MONITOR (@toggle) SITECON HEAVY NO_TEL/@lock-teleport N/A N/A N/A HTML N/A FREE IMMORTAL N/A INHERIT INHERIT JUMP_OK JUMP_OK JUMP_OK KEEPALIVE KEEPALIVE (@toggle) KEEPALIVE N/A KEY KEY LIGHT LIGHT LIGHT LINK_OK LINK_OK LINK_OK LISTEN_PARENT (@admin listen_parents) N/A LOUD NO_OVERRIDE/NO_USELOCK N/A (see @flag) MARKER0-MARKER9 MARKER0-MARKER9 MISTRUST GUILDOBJ/NO_GOBJ/BACKSTAGE/NOBACKSTAGE N/A MONIKER EXTANSI (@toggle) N/A MONITOR MONITOR MONITOR MYOPIC MYOPIC MYOPIC NOACCENTS ACCENTS (@toggle) ACCENTS (Not Needed) (Not Needed) NO_BLEED NOSPOOF NOSPOOF NOSPOOF (See @ns-commands) Auto-Enabled for Wiz+ SPOOF NO_COMMAND NO_COMMAND NO_COMMAND NO_LEAVE @icmd leave/@lock-leave @icmd leave/@lock-leave NO_TEL NO_TEL N/A ON-VACATION MARKER0-MARKER9 VACATION OPAQUE OPAQUE OPAQUE OPEN_OK @lock-openfrom OPEN_OK ORPHAN NOGLOBPARENT (@toggle) N/A N/A PARENT_OK PARENT_OK PUPPET PUPPET PUPPET QUIET QUIET QUIET N/A ROBOT ROBOT ROYALTY GUILDMASTER/ARCHITECT/COUNCILOR ROYALTY (@see @flag) MARKER0-9 STAFF SAFE SAFE SAFE N/A SLAVE SLAVE N/A MONITOR (@toggle) SITEMON STICKY STICKY STICKY N/A SUSPECT SUSPECT TERSE TERSE TERSE TRANSPARENT TRANSPARENT TRANSPARENT UNFINDABLE UNFINDABLE UNFINDABLE N/A N/A UNICODE (See @flag) WANDERER UNINSPECTED VERBOSE VERBOSE VERBOSE VISUAL VISUAL VISUAL WIZARD WIZARD/IMMORTAL WIZARD XTERM256 XTERMCOLOR COLOR256

3.11 Flags and toggles that only exist in RhostMUSH

3.11.1 Flags

ALTQUOTA ANONYMOUS ARCHITECT AUDITORIUM BLIND BOUNCE CLOAK COUNCILOR FUBAR GUILDMASTER GUILDOBJ IMMORTAL INDESTRUCTABLE NO_ANSINAME NO_BACKSTAGE NO_CODE NO_CONNECT NO_EXAMINE NO_FLASH NO_GOBJ NO_MODIFY NO_MOVE NO_NAME NO_OVERRIDE NO_PESTER NO_POSSESS NO_STOP NO_UNDERLINE NO_USELOCK NO_WALLS NO_YELL PRIVATE ROBOT SCLOAK SEE_OEMIT SHOWFAILCMD SIDEFX SPAMMONITOR SPOOF STOP WANDERER ZONECONTENTS ZONEPARENT

3.11.2 Toggles

ATRUSE CHKREALITY CPUTIME EXFULLWIZATTR FORCEHALTED HIDEIDLE IGNOREZONE IMMPROG LOGROOM MAILVALIDATE MAIL_LOCKDOWN MAIL_NOPARSE MAIL_STRIPRETURN MONITOR_AREG MONITOR_BAD MONITOR_CONN MONITOR_CPU MONITOR_DISREASON MONITOR_FAIL MONITOR_SITE MONITOR_STATS MONITOR_TIME MONITOR_USERID MONITOR_VLIMIT MORTALREALITY NODEFAULT NOSHPROG NOZONEPARENT NO_ANSI_EX NO_ANSI_EXIT NO_ANSI_PLAYER NO_ANSI_ROOM NO_ANSI_THING NO_FORMAT NO_TIMESTAMP PAGELOCK PROG PROG_ON_CONNECT SAFELOG SEE_SUSPECT SILENTEFFECT SNUFFDARK ZONECMDCHK ZONE_AUTOADD ZONE_AUTOADDALL

3.12 @power equivalents

@powers Pennmush————————-RhostMUSH————————————————MUX2 Announce FREE_WALL Announce Boot BOOT Boot Builder @quota !WANDERER (flag) Builder CAN_DARK @admin player_dark/@depower dark N/A Can_spoof N/A - Wizard+ auto-spoof N/A Cemit N/A N/A N/A CHOWN_OTHER chown_anything N/A @lock/twink control_all N/A WIZ_WHO expanded_who Chat_Privs N/A comm_all DEBIT STEAL Steal_money Functions (See @lfunctions) Wizard+ only Guest GUEST (flag) Guest HOOK Wizard+ only Wizard+ only Halt HALT_QUEUE/HALT_QUEUE_ALL Halt Hide NOWHO Hide Idle @timeout player to -1 Idle N/A NO_MODIFY (flag) Immutable Link_Anywhere N/A (security risk) N/A Login LOGIN (flag) LOGIN (flag) Long_Fingers LONG_FINGERS Long_fingers MANY_ATTRIBS (@admin vlimit) N/A N/A MONITOR (@toggle) Monitor No_Pay FREE (flag) Free_money No_Quota FREE_QUOTA Free_quota Open_Anywhere N/A (security risk) N/A N/A (Wiz+ Automatic) Pass_locks PICK_DBREFS Wizard+ only N/A PUEBLO_SEND N/A N/A Pemit_All LONG_FINGERS N/A Player_Create PCREATE N/A Poll N/A – Softcode @doing/header Poll N/A PROG (@toggle) Prog Queue SEE_QUEUE/SEE_QUEUE_ALL/HALT_QUEUE/HALT_QUEUE_ALL N/A Quotas CHANGE_QUOTAS N/A SQL_OK N/A N/A Search SEARCH_ANY Search See_All EXAMINE_FULL See_all N/A WHO_UNFIND See_hidden N/A SHUTDOWN Siteadmin See_Queue SEE_QUEUE/SEE_QUEUE_ALL N/A N/A STAT_ANY Stat_any Tport_Anything TEL_ANYTHING Tel_anything Tport_Anywhere TEL_ANYWHERE Tel_anywhere Unkillable NOKILL Unkillable

3.12.1 @power unique to RhostMUSH

Depowers are unique in Rhost and PennMUSH has no equivelant.

Powers that exist in RhostMUSH that have no match in PennMUSH: CHOWN_ME WIZ_WHO NOFORCE FREE_QUOTA JOIN_PLAYER NO_BOOT STAT_ANY WHO_UNFIND SHUTDOWN PURGE CHOWN_ANYWHERE CHOWN_OTHER GRAB_PLAYER SECURITY WRAITH HIDEBIT

3.13 Functions equivalents

Functions Pennmush————————-RhostMUSH————————MUX @@ @@ @@ ABS ABS ABS/IABS ACCENT ACCENT ACCENT ACCNAME CNAME MONIKER ACOS ACOS ACOS ADD ADD ADD AFTER AFTER AFTER ALIAS get(#db/alias)/LISTPROTECT get(#db/alias) ALIGN PRINTF N/A ALLOF OFPARSE N/A ALPHAMAX ALPHAMAX ALPHAMAX ALPHAMIN ALPHAMIN ALPHAMIN AND AND AND/ANDBOOL ANDFLAGS ANDFLAGS ANDFLAGS ANDLFLAGS ANDFLAG N/A ANDLPOWERS @function (softfunctions) N/A ANSI ANSI ANSI APOSS APOSS APOSS ART ART ART ASIN ASIN ASIN ATAN ATAN ATAN ATAN2 ATAN2 ATAN2 N/A ATTRCNT ATTRCNT ATRLOCK HASFLAG(#obj/attr,LOCK) HASFLAG(#obj/attr,LOCK) ATTRIB_SET SET SET BAND MASK BAND BASECONV PACK/UNPACK BASECONV BEEP BEEP BEEP BEFORE BEFORE BEFORE N/A BITTYPE BIT-

TYPE BENCHMARK CPUTIME (@toggle) N/A BNAND MASK BNAND BNOT MASK N/A BOR BOR BOR BOUND BOUND/FBOUND N/A BRACKETS BRACKETS N/A BXOR MASK BXOR CAND CAND CAND/CANDBOOL CAPSTR CAPSTR CAPSTR CASE CASE CASE CASEALL CASEALL N/A CAT CAT CAT CBUFFER N/A N/A CBUFFERADD N/A N/A CDESC N/A N/A CEIL CEIL CEIL CEMIT N/A CEMIT CENTER CENTER CENTER CFLAGS N/A N/A CHANNELS N/A CHANNELS CHECKPASS CHECKPASS N/A CHILDREN CHILDREN CHILDREN N/A N/A CHOOSE CHR CHR CHR CLFLAGS N/A N/A CLOCK N/A N/A CLONE CLONE N/A CMDS CMDS CMDS CMOGRIFIER N/A N/A CMSGS N/A N/A COLORS COLORS N/A N/A N/A COLORDEPTH N/A @function (softfunctions) COLUMNNS N/A N/A COMALIAS COMP COMP COMP N/A N/A COMTITLE CON CON CON COND @function (softfunctions) N/A CONDALL @function (softfunctions) N/A CONFIG CONFIG CONFIG CONN CONN CONN convsecs(get(#db/last)) convsecs(get(#db/last)) CONNLAST N/A N/A CONNLEFT N/A N/A CONNMAX N/A N/A CONNNUM N/A N/A CONNRECORD N/A N/A CONNTOTAL CONTROLS CONTROLS CONTROLS CONVSECS CONVSECS CONVSECS CONVTIME CONVTIME CONVTIME CONVUTCSECS CONVSECS CONVSECS CONVUTCTIME CONVTIME CONVTIME COR COR COR/CORBOOL COS COS COS ALIGN PRINTF CPAD N/A CRC32 CRC32 COWNER N/A N/A CREATE CREATE CREATE CRECALL N/A N/A CSECS N/A N/A CSTATUS N/A N/A CTIME N/A CTIME CTITLE N/A N/A CTU CTU CTU CUSERS N/A N/A CWHO N/A CWHO DEC DEC/XDEC DEC DECODE64 DECODE64 N/A DECOMPOSE TRANSLATE TRANSLATE DECRYPT DECRYPT DECRYPT DEFAULT DEFAULT DEFAULT N/A DESTROY DESTROY DIE DICE DIE DIG DIG CREATE(with 'r') DIGEST DIGEST DIGEST N/A TIMEFMT DIGITTIME DIST2D DIST2D DIST2D DIST3D DIST3D DIST3D @function @function DISTRIBUTE DIV DIV IDIV DOING DOING DOING N/A N/A DUMPING E E E EDEFAULT EDEFAULT EDEFAULT EDIT PEDIT/EDIT EDIT ELEMENT MATCH MATCH ELEMENTS ELEMENTSMUX/ELEMENTS ELEMENTS ELIST ELIST ITEMIZE ELOCK ELOCK ELOCK EMIT EMIT EMIT ENCODE64 ENCODE64 N/A ENCRYPT ENCRYPT ENCRYPT ENDTAG N/A N/A ENTRANCES ENTRANCES ENTRANCES EQ EQ EQ N/A ERROR ERROR ESCAPE ESCAPE ESCAPE ETIME @function (softfunctions) N/A ETIMEFMT TIMEFMT ETIMEFMT EVAL EVAL EVAL EXIT EXIT EXIT N/A EXP EXP EXTRACT EXTRACT EXTRACT %+ %+ FCOUNT %+ %+ FDEPTH FDIV FDIV FDIV FILTER FILTER FILTER FILTERBOOL FILTER FILTERBOOL FINDABLE FINDABLE FINDABLE FIRST FIRST FIRST FIRSTOF OFPARSE N/A FLAGS FLAGS FLAGS FLIP REVERSE REVERSE FLOOR FLOOR FLOOR FLOORDIV FLOORDIV FLOORDIV FMOD FMOD FMOD FN BYPASS N/A FOLD FOLD FOLD FOLDERSTATS FOLDERLIST N/A FOLLOWERS N/A (softcode available) N/A FOLLOWING N/A (softcode available) N/A FOREACH FOREACH FOREACH FRACTION N/A N/A FULLALIAS ALIAS + LISTPROTECT N/A FULLNAME FULLNAME FULLNAME FUNCTIONS LISTFUNCTIONS N/A GET GET GET GETPIDS PIDS N/A GET_EVAL GET_EVAL GET_EVAL GRAB GRAB GRAB GRABALL GRABALL GRABALL GREP GREP GREP GREPI GREP GREPI GT GT GT GTE GTE GTE HASATTR HASATTR HASATTR HASATTRP HASATTRP HASATTRP HASATTRPVAL HASATTRP HASATTRP HASATTRVAL HASATTR HASATTR HASFLAG HASFLAG HASFLAG HASPOWER HASPOWER HASPOWER N/A HASQUOTA HASQUOTA HASTYPE HASTYPE HASTYPE HEIGHT @function (softfunctions) HEIGHT HIDDEN HIDDEN N/A HOME HOME HOME HOST LOOKUP_SITE HOST HTML N/A N/A IBREAK IBREAK N/A IDLE IDLE IDLE IF IFELSE IF IFELSE IFELSE IFELSE ILEV ILEV ILEV INAME NAME NAME INC INC/XINC INC INDEX INDEX INDEX INUM INUM/INUM2 INUM N/A INZONE INZONE IPADDR LOOKUP_SITE N/A ISDAYLIGHT TIMEFMT N/A ISDBREF ISDBREF ISDBREF ISINT ISINT ISINT ISNUM ISNUM ISNUM N/A N/A ISRAT ISOBJID N/A N/A ISREGEXP N/A N/A ISWORD ISWORD ISWORD ITEMIZE ELIST ITEMIZE ITEMS WORDS WORDS ITER ITER ITER ITEXT ITEXT ITEXT LALIGN PRINTF N/A STRMATH LADD LADD N/A LAND LAND LAST LAST LAST N/A LASTCREATE LASTCREATE LATTR LATTR LATTR N/A LATTR LATTRCMDS LATTRP LATTRP LATTRP N/A LCMDS LCMDS LCON LCON LCON LCSTR LCSTR LCSTR LDELETE LDELETE LDELETE LEFT LEFT STRTRUNC LEMIT LEMIT N/A LETQ @function (softfunctions) N/A LEXITS LEXITS LEXITS LFLAGS LFLAGS LFLAGS LINK LINK LINK N/A LIST (like iter()) LIST (like iter()) LINSERT INSERT INSERT LIST LISTPOWERS, FLAGS, etc N/A LISTQ N/A N/A LIT LIT LIT LJUST LJUST LJUST LLOCKFLAGS N/A N/A LLOCKS LOCKS LOCKS LMATH STRFUNC N/A LN LN LN LNUM LNUM/LNUM2 LNUM LOC LOC LOC LOCALIZE LOCALIZE LOCALIZE LOCATE LOCATE LOCATE LOCK LOCK LOCK N/A LOG2FILE LOG LOCKFILTER LOCKCHECK N/A LOCKFLAGS FLAGS FLAGS LOCKOWNER OWNER OWNER LOCKS LOCK LOCK LOG LOG LOG LPARENT PARENTS LPARENT

LPIDS PIDS N/A LPLAYERS LCON LCON N/A LOR LOR LPORTS PORT PORTS ALIGN PRINTF LPAD LPOS LPOS LPOS DIE DICE LRAND N/A LROOMS LROOMS LREPLACE REPLACE REPLACE LSEARCH SEARCH/SEARCHNG SEARCH LSEARCHR revwords(search()) revwords(search()) LSET SET SET LSTATS STATS STATS LT LT LT LTE LTE LTE LTHINGS LCON LCON LVCON LCON + STREVAL at mortal N/A LVEXITS LCON + STREVAL at mortal N/A LVPLAYERS LCON + STREVAL at mortal N/A LVTHINGS LCON + STREVAL at mortal N/A LWHO LWHO LWHO LWHOID N/A N/A MAIL MAILREAD/MAILSEND MAIL MAILDSTATS MAILSIZE/MAILQUOTA MAILSIZE MAILFROM MAILREAD MAILFROM MAILFSTATS FOLDERLIST/FOLDERCURRENT N/A MAILLIST MAILREAD N/A MAILSEND MAILSEND N/A MAILSTATS MAILSIZE/MAILQUOTA MAILSIZE MAILSTATUS MAILSIZE/MAILQUOTA MAILSIZE MAILSUBJECT MAILREAD MAILSUBJ MAILTIME MAILREAD N/A MALIAS MAILREAD N/A MAP MAP MAP MAPSQL N/A N/A MATCH MATCH MATCH MATCHALL MATCHALL MATCHALL MAX MAX MAX MEAN AVG AVG MEDIAN AVG AVG MEMBER MEMBER MEMBER MERGE MERGE MERGE MESSAGE PARSESTR + PEMIT/REMIT N/A MID MID MID MIN MIN MIN MIX MIX MIX MODULO MOD MOD MONEY MONEY MONEY MONIKER CNAME MONIKER N/A N/A MOTD MSECS MODIFYTIME + CONVTIME MTIME + CONVTIME MTIME MODIFYTIME MTIME MUDNAME MUDNAME MUDNAME MUDURL N/A N/A MUL MUL MUL MUNGE MUNGE MUNGE MWHO LWHO + STREVAL at mortal N/A MWHOID N/A N/A NAME NAME NAME NAMEGRAB @function (softfunctions) N/A NAMEGRABALL @function (softfunctions) N/A NAMELIST @function (softfunctions) N/A NAND NAND N/A NATTR ATTRCNT ATTRCNT NATTRP ATTRCNT ATTRCNT NCAND !CAND NOT(CAND()) NCHILDREN CHILDREN CHILDREN NCON WORDS + LCON WORDS + LCON NCOND @function (softfunctions) N/A NCONDALL @function (softfunctions) N/A NCOR !COR NOT(COR()) NEARBY NEARBY NEARBY NEQ NEQ NEQ NEXITS WORDS + LEXITS WORDS + LEXITS NEXT NEXT NEXT NEXTDBREF N/A N/A NLSEARCH WORDS + SEARCH WORDS + SEARCH NMWHO WORDS+LWHO+STREVAL at mort N/A NOR NOR N/A NOT NOT or ! NOT NPLAYERS WORDS + LCON WORDS + LCON NSCEMIT N/A N/A NSEARCH WORDS + SEARCH WORDS + SEARCH NSEMIT EMIT EMIT NSLEMIT LEMIT N/A NSOEMIT OEMIT N/A NSPEMIT PEMIT N/A NSPROMPT N/A (@program?) N/A (@program?) NSREMIT REMIT N/A NSZEMIT ZEMIT N/A NTHINGS WORDS + LCON WORDS + LCON NULL NULL NULL NUM NUM NUM NUMVERSION N/A N/A NVCON WORDS+LCON+STREVAL at mort N/A NVEXITS WORDS+LEXITS+STREVAL at mo N/A NVPLAYERS WORDS+LCON+STREVAL at mort N/A NVTHINGS WORDS+LCON+STREVAL at mort N/A NWHO WORDS + LWHO WORDS + LWHO OBJ OBJ OBJ OBJEVAL OBJEVAL OBJEVAL OBJID N/A N/A OBJMEM SIZE OBJMEM OEMIT OEMIT OEMIT OPEN OPEN N/A OR OR OR/ORBOOL ORD ASC ORD ORDINAL N/A N/A ORFLAGS ORFLAGS ORFLAGS ORLFLAGS ORFLAG N/A ORLPOWERS N/A (easy to @function) N/A OWNER OWNER OWNER PARENT PARENT PARENT PCREATE CREATE CREATE PEMIT PEMIT PEMIT PFUN U + PARENT U + PARENT PI PI PI PIDINFO PID N/A PLAYER before(grab(lwho(1),*:%0),:) N/A PLAYERMEM SIZE PLAYMEM PMATCH PMATCH PMATCH POLL DOING POLL PORTS PORT PORTS POS POS POS POSS POSS POSS POWER POWER POWER POWERS LPOWERS POWERS PROMPT N/A (@program?) N/A (@program?) PUEBLO N/A N/A QUOTA QUOTA N/A R R R RAND RAND RAND RANDWORD PICKRAND PICKRAND RECV CHARIN N/A REGEDIT REGEDIT N/A REGEDITALL REGEDITALL N/A REGEDITALLI REGEDITALLI N/A REGEDITI REGEDITI N/A REGISTERS N/A N/A REGLATTR LATTR N/A REGLATTRP LATTRP N/A REGLMATCH REGLMATCH N/A REGLMATCHALL REGLMATCHALL N/A REGLMATCHALLI REGLMATCHALLI N/A REGLMATCHI REGLMATCHI N/A REGMATCH REGMATCH REGMATCH REGMATCHI REGMATCHI REGMATCHI REGNATTR WORDS + ATTR N/A REGNATTRP WORDS + ATTRP N/A REGRAB REGRAB REGRAB REGRABALL REGRABALL REGRABALL REGRABALLI REGRABALLI REGRABALLI REGRABI REGRABI REGRABI REGREP REGREP N/A REGREPI REGREPI N/A REGXATTR ATTR N/A REGXATTRP ATTRP N/A REMAINDER REMAINDER REMAINDER REMIT REMIT REMIT REMOVE REMOVE REMOVE RENDER N/A N/A REPEAT REPEAT REPEAT REST REST REST RESTARTS N/A RESTARTS RESTARTTIME REBOOTTIME RESTARTTIME CONVTIME(RESTARTTIME)) CONVTIME(REBOOTTIME()) RESTARTSECS RESWITCH RESWITCH N/A RESWITCHALL RESWITCHALL N/A RESWITCHALLI RESWITCHALLI N/A RESWITCHI RESWITCHI N/A REVWORDS REVWORDS REVWORDS RIGHT RIGHT RIGHT RJUST RJUST RJUST RLOC RLOC RLOC N/A ROMAN ROMAN RNUM RNUM N/A ROOM ROOM ROOM ROOT N/A N/A ROUND ROUND ROUND ALIGN PRINTF RPAD S S S SCAN N/A N/A SCRAMBLE SCRAMBLE

SCRAMBLE SECS SECS SECS SECURE SECURE/SECUREX SECURE SENT CHAROUT N/A SET SET SET SETDIFF SETDIFF SETDIFF SETINTER SETINTER SETINTER SETQ SETQ SETQ SETR SETR SETR SETUNION SETUNION SETUNION SHA0 DIGEST DIGEST DIGEST DIGEST SHA1 SHL SHL SHL SHR SHR SHR SHUFFLE SHUFFLE SHUFFLE SIGN NCOMP(%0,0) SIGN SIN SIN SIN SLEV N/A N/A @function @function SITEINFO SORT SORT SORT SORTBY SORTBY SORTBY SORTKEY @function (softfunctions) N/A SOUNDEX SOUNDEX N/A SOUNDSLIKE SOUNDXLIKE N/A SPACE SPACE SPACE SPEAK PARSESTR N/A SPEAKPENN PARSESTR N/A SPELLNUM SPELLNUM SPELLNUM SPLICE SPLICE SPLICE SQL (if MYSQL enabled) N/A (ASYNC db) SQLESCAPE (if MYSQL enabled) N/A (ASYNC db) SQRT SQRT SQRT SQUISH SQUISH SQUISH SSL N/A N/A STARTTIME STARTTIME STARTTIME CONVTIME(STARTTIME)) CONVTIME(STARTTIME()) STARTSECS N/A STATS STATS STDDEV AVG AVG STEP STEP STEP STEXT N/A N/A STRALLOF OFPARSE N/A STRCAT STRCAT STRCAT N/A STRIP STRIP STRDELETE CREPLACE/DELETE DELETE STRFIRSTOF OFPARSE N/A STRINGSECS @function (softfunctions) N/A STRINSERT CREPLACE N/A STRIPACCENTS STRIPACCENTS STRIPACCENTS STRIPANSI STRIPANSI STRIPANSI STRLEN STRLEN STRLEN STRMATCH STRMATCH STRMATCH N/A STRLENRAW STRMEM STRREPLACE CREPLACE/REPLACE REPLACE SUB SUB SUB N/A ESCAPEX SUBEVAL SUBJ SUBJ SUBJ N/A N/A SUCCESSES SWITCH SWITCH SWITCH SWITCHALL SWITCHALL N/A T T T TABLE @function (softfunctions) TABLE TAG N/A N/A TAGWRAP N/A N/A TAN TAN TAN TEL TEL TEL TERMINFO N/A TERMINFO TESTLOCK LOCKCHECK N/A TEXTENTRIES WORDS + TEXTFILE WORDS + TEXTFILE TEXTFILE TEXTFILE TEXTFILE TIME TIME TIME TIMEFMT PTIMEFMT TIMEFMT TIMESTRING SINGLETIME/TIMEFMT SINGLETIME TR TR TR TRIM TRIM TRIM TRIMPENN TRIM TRIM TRIMTINY TRIM TRIM N/A N/A TRIGGER TRUNC TRUNC TRUNC TYPE TYPE TYPE UCSTR UCSTR UCSTR UDEFAULT UDEFAULT UDEFAULT UFUN U U ULAMBDA U + #lambda N/A ULDEFAULT ULDEFAULT N/A ULOCAL ULOCAL ULOCAL UNIQUE LISTDIFF/LISTUNION/LISTINTER N/A UNSETQ N/A N/A UPTIME N/A N/A UTCTIME TIME TIME V V V VADD VADD VADD VALID VALID VALID VCROSS VCROSS VCROSS VDIM VDIM VDIM VDOT VDOT VDOT VERSION VERSION VERSION VISIBLE VISIBLE VISIBLE VMAG VMAG VMAG VMAX SORTLIST N/A VMIN SORTLIST N/A VMUL VMUL VMUL VSUB VSUB VSUB VUNIT VUNIT VUNIT WHERE WHERE WHERE WIDTH @function (softfunctions) WIDTH WILDGREP GREP N/A WILDGREPI GREPI N/A WIPE WIPE WIPE WORDPOS WORDPOS WORDPOS WORDS WORDS/MWORDS WORDS WRAP WRAP WRAP N/A MODIFYTIME WRITETIME XATTR ATTR N/A XATTRP ATTRP N/A XCON XCON N/A XEXITS LEXITS + EXTRACT N/A XGET XGET XGET XMWHO LWHO+EXTRACT+STREVAL at mor N/A XMWHOID N/A N/A XOR XOR XOR XPLAYERS XCON N/A XTHINGS XCON N/A XVCON XCON + STREVAL at mortal N/A XVEXITS LEXITS + STREVAL at mortal N/A XVPLAYERS XCON + STREVAL at mortal N/A XVTHINGS XCON + STREVAL at mortal N/A XWHO LWHO + EXTRACT LWHO + EXTRACT XWHOID N/A N/A ZEMIT ZEMIT N/A ZFUN ZFUN ZFUN ZMWHO ZWHO + STREVAL at mortal N/A ZONE LZONE ZONE ZWHO ZWHO ZWHO

3.13.1 Functions that only exist in RhostMUSH

Functions that exist in Rhost that do not have a match in PennMUSH: AIINDEX AINDEX ANDCHR ARRAY ATTRCNT BETWEEN BITTYPE CANSEE CAPLIST CHKGARBAGE CHKREALITY CHKTRACE CHOMP CITER CLOAK CLUSTER_ADD CLUSTER_ATTRCNT CLUSTER_DEFAULT CLUSTER_EDEFAULT CLUSTER_FLAGS CLUSTER_GET CLUSTER_GET_EVAL CLUSTER_GREP CLUSTER_HASATTR CLUSTER_HASFLAG CLUSTER_LATTR CLUSTER_REGREP CLUSTER_REGREPI CLUSTER_SET CLUSTER_STATS CLUSTER_U CLUSTER_U2 CLUSTER_U2DEFAULT CLUSTER_U2LDEFAULT CLUSTER_U2LOCAL CLUSTER_UDEFAULT CLUSTER_UEVAL CLUSTER_ULDEFAULT CLUSTER_ULOCAL CLUSTER_VATTRCNT CLUSTER_WIPE CLUSTER_XGET COLUMNS COSH COUNTSPECIAL CRC32 DELEXTRACT DESTROY EDITANSI EE ERROR EXP FBETWEEN FBOUND GARBLE GLOBALROOM GUILD HASDEPOWER HASQUOTA HASRXLEVEL HASTOGGLE HASTXLEVEL INPROGRAM INZONE ISALNUM ISALPHA ISCLUSTER ISDIGIT ISHIDDEN ISLOWER ISPUNCT ISSPACE ISUPPER ISXDIGIT KEEPFLAGS KEEPTYPE LAND LAVG LCMDS LDEPOWERS LISTMATCH LISTNEWSGROUPS LISTRLEVELS LISTTOGGLES LLOC LMAX LMIN LMUL LNOR LOCALFUNC LOCKDECODE LOCKENCODE LOGSTATUS LOGTOFILE LOR LRAND LROOMS LTOGGLES LXNOR LXOR MONEYNAME MOON

MOVE NAMEQ NOSTR NOTCHR NSLOOKUP ORCHR PARENMATCH PFIND PGREP POWER10 PRIVATIZE PROGRAMMER PUSHREGS RACE RANDMATCH RANDPOS REGEDITALLILIT REGEDITALLLIT REGEDITILIT REGEDITLIT REGNUMMATCH REGNUMMATCHI REMFLAGS REMTYPE ROMAN ROTL ROTR RSET RXLEVEL SAFEBUFF SEES SETQMATCH SHIFT SINH SORTLISAT STR STRDISTANCE STREQ STREVAL STRFUNC STRIP STRLENRAW STRLENVIS STRMATH SUBNETMATCH TANH TOGGLE TOTCMDS TRACE TXLEVEL UEVAL WHILE WILDMATCH WRAPCOLUMNS WRITABLE XNOR XORCHR XORFLAG ZFUNDEFAULT ZFUNEVAL ZFUNLDEFAULT ZFUNLOCAL

3.14 What may need to be modified to get softcode from PennMUSH, TinyMUSH2, TinyMUSH3, or MUX2 to work on Rhost

RhostMUSH, for the most part, will work out of the box with most softcode gotten from other codebases. There are, however, exceptions. Most of these exceptions will be minor code differences between how ANSI is processed, the variences of arguments or switches to commands or functions, or required flags.

Most changes will revolve around the ones listed in this document.

3.14.1 Problematic functions between codebases

lsearch()/search(), align()/printf(), *attrval()

3.14.2 Problematic features between codebases

named variables for regexp patterns in $commands are not supported. @aliases on non-players are not supported. Frankly I find them redundant.

3.14.3 Problematic commands

@mapsql, hardcoded required comssytem commands (some are redundant)

3.14.4 SIDEFX flag

Anything that uses sideeffects –DIRECTLY– requires this flag. Sideeffects are like set(), pemit(), and so forth. list(), while a side-effect, does not require this flag as it is considered passive and safe.

3.14.5 Variable exits

Rhost handles them slightly different. You do not link exits to #-4. That's an invalid destination. I always found it, frankly, stupid to save any data in the database that was literally invalid. So, you link the exit as you normally would, then @toggle the exit variable. At that point you use @exitto like you would any other codebase.

3.14.6 Zones

Zones actually can work near exactly as you would expect them to work on TinyMUSH, MUX, or Penn. Either at once or at different times. We recognize multiple zones, zone masters, zone inheritance, zone parenting, zone command processing, and the ability to bypass zones entirely. There's a ton of flexbility with this. However, the syntax for adding/removing zones is different so the commands will have to be ported to Rhost.

3.14.7 @powers

Powers work a bit differently in Rhost and they're named differently, which should not be that big a surprise as they're different between all the codebases anyway. The big difference is our powers are tiered, meaning the can be limited or grown to a given bitlevel and are not just toggle powers like the other codebases. We also have @depower that is the anti-thesis of @power

3.14.8 Attribute length

While we have 64 character attribute capabilities like most other codebases, PennMUSH allows 1024 attribute length attributes. Why you need one that long boggles the mind, but if you do use attribs that long you need to make sure they are cut down to the proper length.

3.14.9 Attribute contents

You'll be happy to know that Rhost allows upwards to 64,000 bytes of data to be assigned an LBUF. We strongly recommand to cap at 32,000 however as the various TCP socket protocols play nicer with that value.

3.14.10 256 color

Yup! We got it.

3.14.11 Unicode/UTF8

Yup! We got this too. Not quiet yet in the main branch, but download Kage's branch, you won't be dissapointed. We will have UTF8 in Rhost 4.0 when released.

3.14.12 Attributes per object

This is configurable with the VLIMIT @admin command, however, it is absolutely hard-limited at 10000 maximum. This is to avoid any DoS type situation and because frankly there should never be a reason to exceed that. If you need more, use @clusters.

3.14.13 Destroying

@nuke only works on players. @destroy works on non-players. Never the two will meet. We also have a built in recycle bin meaning anything destroyed will not be automatically recycled. If you want it recycled, you have to @purge it. Yes, if you use Myrddin's CRON, it has a built in entry to auto-purge anything older than 30 days. This also means you can on-line recover anything destroyed before that 30 days. Groovy, eh?

3.14.14 object id's

Yup, we got them. Even in searches, and, well, everything.

3.14.15 lsearch() and search()

lsearch() in Penn is not syntacically similar to non-Penn search(). This will have to be altered. In addition, search() in non-penn games have to have special consideration for escaping out the evaled args.

3.14.16 @locks can be different

We have many more lock capabilities and options so this should be a non-issue.

3.14.17 Customer user-locks

We do not have custom user-locks like Penn. We do, however, have the way to set encapsulated lock data into an attribute to fetch and compare against which I find more useful and far more flexible. See: lockencode(), lockdecode(), and lockcheck()

3.14.18 Attribute trees

Unlike Penn, we don't really have attribute trees. We do support the basic capabilities of it for compatibility if you load in softcode that uses it, but it doesn't have the advanced features of attribute trees. Please see 'help attribute tree' for more information.

3.14.19 Prefix permission locking

We do allow prefix permission locking, and some very advanced abilities of it. Please see wizhelp on @aflags for more information. - wizhelp @aflags - wizhelp atrperms_max - wizhelp atrlock - wizhelp atrperms

3.14.20 align() and printf()

We do not have align(). Most of the code that uses align() will have to be converted to our printf() (which is compatible but has different syntax)

3.14.21 MySQL

While we support MySQL, we do not have an async method like MUX2. This is just not possible, sorry.

3.14.22 Mail System

There are mail wrappers to mimic MUX/TM3 and Penn mail systems.

3.14.23 Comsystem

The softcoded comsystem mimics MUX/TM3 and Penn's comsystem.

3.14.24 Various Functions

There is a 'softcode.minmax' in the Mushcode directory that loads up a slew of @function wrappers that will emulate various functions that MUX, Penn, or TM3 has. We have the functionality for nearly all of them, but either our functions have different syntax, or we have different named functions that duplicate the functionality. It would be far better to recode it to use the native functions, but the @function wrappers are there for lazyness :)

3.14.25 Empty Attributes

Penn allows you to have empty attributes. Non-penn codebases do not. Thus, hasattrval and the like are not needed and should likely just point to hasattr instead.

3.14.26 Player Stats

MUX has some built in ways for player stats. We do as well but they're either done via functions or attribute contents. Code that requires this will have to be recoded.

3.14.27 Percent Substitutions

Some percent substitutions may differ between codebases. Luckily, Rhost allows the ability to remap or creaete new ones if this is a problem.

3.14.28 Switches

Some switches may not exist in Rhost that do in other codebases, in such a case, Rhost does allow the ability to @hook a command to define your own softcoded switch to a hardcoded command and work around the limitation.

3.14.29 Adding Flags

Some flags may be missing. If it's a dummy flag, feel free to use the marker flags MARKER0 to MARKER9 to set them. If it's an existing flag that does similar features, feel free to flag_alias it or just flag_name it to the other name if you want.

3.14.30 Aliases

Multiple aliases are supported via @protect.

3.15 Things other mushes can do that Rhost can not and how to emulate it

3.15.1 PennMUSH

- Attribute trees. These are emulated as a base set and can be duplicated enough to at least port code.
- lsearch() will have to be recoded to search()
- align() will have to be recoded to printf()
- Penn allows empty attributes. Rhost does not. Work will have to be done to take this into consideration.
- All *val() functions in Penn that are used will have to be remapped to a non-*val() function. It should be as simple as function_alias to the non-*val. Example:

```
@admin function_alias=hasattrval hasattr
```

- Penn's hardcoded comsystem is emulated with the softcode comsys
- Penn's @mail system is workable with mail wrappers
- Pueblo is not supported.
- json is not supported.
- ssl is not natively supported (yet).
- This uses @shutdown/restart, Rhost uses @reboot

3.15.2 MUX

- Mux has an async mysql database engine. This is not possible with Rhost. You'll have to use the sync method instead.
- UTF8 is supported but internally passed differently.
- Mux's hardcoded comsystem is emulated with the softcode comsys
- MUX's @mail system is workable with mail wrappers
- Pueblo is not supported.
- This uses @restart, Rhost uses @reboot

CHAPTER

FOUR

DATABASE

4.1 Loading an existing Database

To load in a previous database, you run the db_load script.

From the game directory you would type:

```
./db_load data/netrhost.gdbm yourflatfilehere data/netrhost.db.new
```

Note: You may also do: ./Startmush Then you just follow the prompts to load in your flatfile there.

If you wish to have #1's password reset to 'Nyctasia' please add this to the bottom of your netrhost.conf file:

```
newpass_god 777
```

The caveat is that you must not have any netrhost.db* or netrhost.gdbm* files in your data directory prior to loading it in. It'll error out if previous files exist. So you will need to move all files that start with netrhost.db* and all files that start with netrhost.gdbm* to another directory.

Your flatfile tends to be named 'netrhost.db.flat' which is in your data directory. You can, however, name your flatfile anything you want and have it in any directory you want.

To make a flatfile in game, you just issue @dump/flat. You can specify a filename after it, otherwise it assumes the name 'netrhost.db.flat'.

4.2 Converting a flatfile database for use in RhostMUSH

In the ~/Server/convert directory there is a script called 'doconvert.sh'

This script will convert most flatfiles from existing mush engines to RhostMUSH. The exception is PennMUSH 1.8.0 and later. For this there is a BETA converter penn18x_converter.tgz. This is proven to work, most of the time, with codebases between 1.8.0 and 1.8.2. It has not been fully vetted with the latest PennMUSH databases. Our apologies.

To convert a non-pennmush game (or a pennmush 1.7.4 or earlier), you first need a valid flatfile of the game you're wanting to convert. Please refer to the documentatation of the mush engine in question (MUX, Penn 1.7, TM2/3) on how to do this. Once you have it type::

```
./doconvert.sh FLATFILETOCONVERT FLATFILEOUTPUT
```

In this instance, FLATFILETOCONVERT will be the filename (with full path) to the flatfile you are wishing to convert.

The FLATFILEOUTPUT is anyfilename you wish to name the RhostMUSH converted flatfile. I suggest netrhost_converted.db.flat so you know by the name what it is.

Follow what it asks for and just let it do its thing.

4.3 Note about Compiling

If you are importing a MUX2 flatfile, make ABSOLUTELY SURE that you select mux passwords as a compatibility option, or you will NOT BE ABLE to log in to players as the password will not be recognizeable.

Make sure to keep QDBM selected as it's a much more stable database engine that does not have attribute limit restrictions like GDBM does.

If you are converting from a Penn, TinyMUSH, or MUX database, make sure you drill down into the LBUF section and select, at minimum, 8K lbufs. You likely want that anyway as it gives you far more room for attribute content storage.

You can go up to 32K safely. While 64k is safe and does work, there are issues with networking and older routers that use a 32K TCP buffer size that can at times cut off the data as overflow resulting in output to the end-point players not receiving their data. So it is strongly recommended not to go above 32K in lbuffer size.

Go ahead and select 64 char attributes. It allows you to have 64 characters for attribute names. It's handy to have.

If you wish at this point to set up mysql and/or sqlite, you may do so. Yes, you can use them in parallel without any issue.

CHAPTER

FIVE

GETTIN STARTED

5.1 What to ype to get the basics running if you did not choose a pre-existing flatfile

If you decided to get a bare-bone configuration, you will find your mush has just two things. The #1 (God) player and the starting room #0. That's it.

5.1.1 Login to #1 from the connect screen

Nyctasia is the default password:

```
co #1 Nyctasia
```

5.1.2 Change #1's password to something you'll remember but is hard to guess

Note: yourpasswordgoeshere can be any password you choose. Choose well:

```
@password Nyctasia=YOURPASSWORDGOESHERE
```

5.1.3 Master Room

At this point you should create your master room:

```
@dig Master Room
```

Note: Reason: You need a master room to contain global $commands for players ^listens are not global for intentional reasons. It's far too much overhead for far too minimal gains that few people need or use.

5.1.4 Flag and protect Master Room

It will return a dbref#, it should be #2 if you've not created anything else:

```
@set #2=safe ind halt float
```

5.1.5 Player Holder Characters

Feel free to change the password to what you want

Note: Reason: You will want to chown global room or global areas to a given bitlevel and a method to keep organized. Note: wizhelp control will give you a complete breakdown of what each bit can do.

Immortal Holder

```
@pcreate ImmHolder=abc123
@set *Immholder=no_connect !wanderer immortal
@badsite *immholder=*
```

Royalty/Wizard Holder

```
@pcreate WizHolder=abc123
@set *wizholder=no_connect !wanderer royalty
@badsite *wizholder=*
```

Councilor/Admin Holder

```
@pcreate AdminHolder=abc123
@set *adminholder=no_connect !wanderer councilor
@badsite *adminholder=*
```

Architect/Staff Holder

```
@pcreate StaffHolder=abc123
@set *staffholder=no_connect !wanderer architect
@badsite *staffholder=*
```

Guildmaster/Lead Holder

```
@pcreate GuildHolder=abc123
@set *guildholder=no_connect !wanderer guildmaster
@badsite *guildholder=*
```

5.1.6 Chown #0 (The starting room) and #2 (The Master room) to immholder

Note: #0 you can chown to a different bitlevel if you want, but the master room should always be owned by an immortal

```
@chown/preserve #0=*immholder
@chown/preserve #2=*immholder
```

5.1.7 Create yourself your own immortal player then log off #1 and into this immortal player

Note: Pick what you want for playername and playerpassword

```
@pcreate PLAYERNAME=PLAYERPASSWORD
@set *playername=!wanderer immortal
```

5.1.8 Log out of #1 and log into your immortal player

Note: Use the playername and password you created in the step before

```
LOGOUT
co PLAYERNAME PLAYERPASSWORD
```

5.1.9 Create your guest characters

Note: Feel free to change the description if you want

```
@dolist lnum(1,10)={@pcreate Guest##=guest;@set *Guest##=guest;@adisconnect *Guest#
↪#=home;@lock *Guest##=*Guest##;@desc *Guest##=A Stranger in a strange land.}
```

5.1.10 Dig a closet to store important objects but non-master room

Note: name it anything you want, just remember it.

```
@dig Closet
```

5.1.11 Set the flags on the closet and ownership of it

Note: Use the dbref# that it returned when digging the closet and not #123

```
@set #123=inh safe ind float
@chown/pres #123=*immholder
```

5.1.12 Create an Admin object for future ease of customization

```
@create AdminObject
```

5.1.13 Set the flags on the admin object and ownership and location

Note: this object must be immortal owned. Use the dbref# returned previously instead of #123

Note: Use the closet dbref# instead of #234

```
@set AdminObject=halt safe ind
@chown/pres #123=*immholder
@tel adminobject=#234
```

5.1.14 Add admin object to configuration

Modify the netrhost.conf file with the following line at the bottom after the line '# define local alises here' where you replace 123 with the dbref# of the admin object that you made:

```
admin_object 123
```

5.1.15 Reboot your mush to load up the change for the admin object

```
@reboot
```

5.1.16 Do @admin/list to see if it shows the admin object

Note: do wizhelp @admin for more information on how to use this

```
@admin/list
```

5.1.17 Load in all the various softcode that you want

This is client dependant based on whatever method it uses to load softcode.

Myrddin MushCron

Load in the Myrddin Mush Cron. It's a very handy piece of software and strongly suggested to do so. You can find this in the 'Mushcode' directory off the main Rhost directory. Filename:

```
~/Rhost/Mushcode/MyrddinCRON
```

Note: The globalroom() function returns the dbref# of the master room. Handy!

```
@chown/preserve the myrddin mush cron to immholder, then move to maste room.
@chown/pres Myrddin=*Immholder
@tel Myrddin=#234 (where #234 is the dbref# of your code closet)
```

AshCom

Load in default softcoded comsystem. This is PennMUSH and MUX/TM3 compatible. Filename:

```
~/Rhost/Mushcode/comsys
```

Chown the Comsystem and everything inside it to immholder:

```
@chown/pres ChanSys=*Immholder
@dolist lcon(chansys)=@chown/pres ##=*immholder
@tel Chansys=globalroom()
```

Mail Wrappers

Load in mail wrappers if you want MUX/TM3 and/or Penn mail wrapping. Filename (MUX/TM3):

```
~/Rhost/Mushcode/mailwrappers/muxmail.wrap
```

Filename (Penn):

```
~/Rhost/Mushcode/mailwrappers/pennmail.wrap
```

Chown to immholder:

```
@chown/pres MUX=*Immholder
@chown/pres Penn=*Immholder
@tel/list mux penn=globalroom()
```

Myrddin BBS

Load in Myrddin's BBS Filename:

```
~/Rhost/Mushcode/MyrddinBBS
```

Chown to immholder and the contents of it as well:

```
@chown/pres Myrddin=*Immholder
@dolist lcon(myrddin)=@chown/pres ##=*immholder
@tel myrddin=globalroom()
```

Other Mushcode

There's other code in the Mushcode directory that you are welcome to install. You would follow similar procedures for loading it in, chowning it, and moving to the master room as you did above.

Likewise, any softcode you find on the internet or on other mushes should be portable to RhostMUSH with little to no changes depending on the complexity of the code in question.

5.2 Minimal DB instructions

The 'retired' directory has older image files if you're interested

Please use the netrhost.conf file with the database as they're linked.

The flatfile must be loaded in as a new db

This is a minimal db with basic 'features' built in.

Copy the txt files into the Rhost's txt directory off game:

```
cp txt/* ~/Rhost/Server/game/txt
```

mkindx the files (substitute FILENAME with the filename):

```
cd ~/Rhost/Server/game/txt
../mkindx FILENAME.txt FILENAME.indx
```

5.2.1 Startup Steps

1. Using the Startmush utility for the first time, select the load db method

— or —

1. copy the netrhost.conf file into the games directory
2. make any relevant changes you wish
3. db_load the flatfile
    1. go into the game directory
    2. type:

```
./db_load data/netrhost.gdbm ../minimal-DBs/minimal_db/netrhost.db.flat data/
↪netrhost.db.new
```

4. Startmush as expected

5.3 Ambrosia's Minimal Rhost DB

5.3.1 Version: 1.0.5 2020-01-31

Version history:

1.0.0

- Initial database setup.

1.0.1

- Small fixes of objid(), isstaff() and bccheck() permissions and handling.
- bittype() access lowered to Architect level
- NO_CODE flag made visual to Architect

1.0.2

- Several convenience changes and fixes: _ Attributes moved to @aflags system, allowing Architects to set, Guildmasters to see them.
- @flagdef lowered to Royalty level. @quota/max, @quota/unlock and @convert moved to Architect level.
- NO_CODE flag made settable/unsettable by Architects.
- Fixed typo in conf file: ifselse -> ifelse
- Switched _Attributes to use the @aflags system See: Guildmaster Set: Architect

1.0.3

- Removed @flagdefs from in-game softcode, converted to flag_access_* config options
- Lowered mailstatus() access to architect.

1.0.4

- Changed softcoded objid() to tag(), due to Rhost's new hardcoded objid() which does perform a different functionality.

- Added more staff recommendations to this file.
- Added Reality TXLevel 'Admin' to all objects in the db except #1

1.0.5

- Replaced softcoded tag() system with Rhost's new hardcoded @tag/tag() system. All previous tags are set on the database. The Tag Object was removed.
- @function startup on BC-Admin-Royalty fixed - @wait 1 workaround for Tags in place.
- Places System @startup integrated into BC-Admin-Royalty's @startup
- Made @dump and @dump/flat available to Councilors in netrhost.conf

1.0.6

- A small typo fix in netrhost.conf. float_preciiosn -> precision and functions_max -> function_max. Thanks to Bobbi@COH

5.3.2 AmbrosiaDB Introduction

Note: READ THIS DOCUMENT CAREFULLY!

Greetings,

This minimal Rhost DB was made with a secure setup, and as a good base to start a new game off in mind.

5.3.3 AmbrosiaDB Features

AmbrosiaDB Configuration

- Limbo, Master Room and Auxiliary room.
- BC-Admin-<bitlevel> characters set up for each bitlevel to own global and data objects, and inherit to.
- BC-Admin-Mortal is @powered EXAMINE_ALL(Guildmaster), NOFORCE(Architect) and LONG_FINGERS.
- @startup on BC-Admin-Immortal lowers DARK flag access to Councilor level, and NO_CODE visual access to Architect level.
- Global Command objects inheriting from each bitlevel, with a separate staff-only object for each level.
- Global Function objects inheriting from each bitlevel.
- local Function objects inheriting from each bitlevel.
- @function and @hook access lowered to Royalty level to remove immediate need for Immortals or actual Immortal code.
- @rxlevel, @txlevel, bittype() access lowered to Architect level to remove immediate need for Royalty in many cases.
- @startup on BC-Admin-Royalty to automatically load @hooks and @functions from the Global Function objects, based on attribute naming.
- Misc Data object to hold general data, like Staff lists etc.
- Reality levels 'Real' and 'Admin'.

Note: All created items and players by default are in Recieve-Level 'Real' and Transmit-Levels 'Real' and 'Admin'.

- All globals, Master Room, BCs-*, and other staff/code-related objects currently have only 'Admin' as their Transmit-Level. This does not prevent them fromi working properly. The only exception is #1, who has empty reality levels.
- The supplied netrhost.conf offers a secure setup of options, allows Royalty to use @hook and @function, and also sets the function_access of several functions to !no_code, which allows NO_CODE players to use the comsys properly.

Note: IT IS HIGHLY RECOMMENDED to use this .conf as a base for this DB. The 'Port' configuration parameter is XXXX'd out. Set it first before starting your game.

- All existing objects have been @set SAFE and INDESTRUCTABLE.
- All existing objects have a paranoid series of @locks set on themselves.
- The +supersafe command is provided on #1 as an example of what was used to easily set this on objects.
- Players are @set NO_CODE and WANDERER by default. They get 40 credits on creation, and a 1-credit-per-day paycheck.
- All *mit sideeffects, as well as set(), create() and list() are enabled. The latter three are necessary for the Comsystem. The rest of sideeffects are disabled completely.
- Flashing ansi is disabled.
- _Attributes are settable by Architects, and seeable by Guildmasters. Read: Still invisible and unsettable by mortals.
- Architects can set up, unlock, and change alternate @quota on players.
- Architects can set/unset the NO_CODE flag.
- Guildmasters can see _Attributes
- Architects can set _Attributes

AmbeosiaDB Softcode

- Set-up compatibility SoftFunctions and @hook object.
- Set up Comsys 1.0.9b at Architect level. (+bbhelp command)
- Set up Myrddin +bboard at Architect level.
- Set up Myrddon Cron at Architect level.
- Anomaly Jobs (+jhelp)
- SGP Places & Mutter
- Set up Penn-style follow.
- Set up @scan (Up to architect-level items).
- Set up Penn- and Mux compatibility Mailwrappers. (phelp and mhelp commands)
- help .txt files and .indx files for the above.
- @dynhelp access lowered to architect to call above helpfiles.

AmbrosiaDB Functions

- isstaff() - Softcoded function that returns '1' if its argument matches a #dbref in the 'isstaff' attribute on the Misc Data object.
- bccheck() - Softcoded function that returns '1' if its argument matches a #dbref in the 'bcs' attribute on the Misc Data object.
- width() - Softcoded function that returns 78 for now. For cross-MU* compatibility.
- pass() - Softcoded function that takes a number as its parameter, and return a random string of that length. Perfect for setting random passwords.
- cmdmessage() - Softcoded function that takes two strings as its parameters. Returns '<< STRING1 >> String2'. The <<>> part is highlighted red. Good for all kinds of messages sent by game commands.
- header() - Highly versatile, and a buffer-saving alternative to using printf() for centering with ansi borders. HIGHLY recommended to use instead of printf() for such things.

```
header(text,width,filler,fillercolor,outerpadding,paddingcolor,
       leftinnerpadding,leftinnercolor,rightinnerpadding,rightinnercolor)
  text - Text to center
  width - Width of the header, defaults to width()
  filler - Character(s) to draw the line with. Default: =
  fillercolor - ansicode to color the line with
  outerpadding - Characters to frame the outer ends of the line with.
  paddingcolor - ansicode to color the outer characters with
  leftinnerpadding - characters to put at the left side of <text>
  leftinnercolor - ansicode to color the leftside characters with
  rightinnerpadding - characters to put at the right side of <text>
  rightinnercolor - ansicode to color the rightside characters with
```

Note: ALL of header()'s parameters are optional. By default, header() simply draws a 78-char wide line of ='s. Simply leave parameters empty if you want to set one of the latter parameters.

5.3.4 AmbrosiaDB Bitlevels

The whole DB is highly geared for a low-bitlevel setup. I am a strong believer in least-privileges-needed to do the job. Bittypes and powers are tools to do that job, not badges of friendship or trust that get tossed about.

Note: Here is my suggested list of powers and bittypes for staffers.

AmbrosiaDB Storytellers

@powered TEL_ANYWHERE, JOIN_PLAYER and GRAB_PLAYER on Guildmaster level.

AmbrosiaDB Builder-BCs

Mortal, with @quota and money for their job. There should be one shared BC for each area of the game, like BC-Houston. Special Rooms, items or exits that require privilegs to examine or @tel a player should belong to a BC-Houston-Powered that is @powered EXAMINE_ALL, LONG_FINGERS And TEL_ANYTHING on Guildmaster level. If the object actually needs to modify a player directly, have it use a restricted staff Global, or if you absolutely must, make a BC-Houston-Admin and @set it Architect. Do not give normal builders access to it, only @chown things to it and @set them inherit after review.

Both the -powered and -admin BCs can have random passwords and be @set NO_CONNECT.

AmbrosiaDB Building Head

@set Guildmaster, powered TEL_ANYWHERE, TEL_ANYTHING and optionally CHOWN_OTHER on Guildmaster level. Mind that the latter technically allows them to @chown anything guildmaster- and lower-owned in the master and auxiliary rooms. However, it allows the Building Head to @chown items between BCs- and to the BC-<location>-powered.

AmbrosiaDB Enforcers

As Storyteller above, plus being @powered Security at Guildmaster level, in order to handle problem players.

Optionally always given to Storytellers.

AmbrosiaDB Coders

@set Architect

AmbrosiaDB Head Coder

Always trust your head coder. @set Architect for the everyday bit. Give access to a maintenance Councilor bit for special code projects. Finished code along those lines should get @chowned to the bc-admin-councilor.

If you as the MU* Head(s) don't know Softcode well, or want to leave anything Code to your Head Coder, also give them optional access to a maintenance Royalty bit in order to properly code banning/blacklisting +commands and other rare code that requires Royalty powers. Again, chown finished code to bc-admin-royalty.

AmbrosiaDB MU* Head(s)

You're the boss(es). But please use an Architect bit for your everyday things. Keep Immortal to yourself. Keep #1 to yourself. And seriously avoid using either of them except for creating a Royalty bit or doing intricate DB maintenance.

AmbrosiaDB Site Admins

They already have more powers than any in-game bit can ever have ;)

Depending on actual involvement with your game, their abilities in-game can range from merely being @powered free_wall for notifying players of downtimes and/or being @powered shutdown in order to shut down the game for maintenance, up to being the only person with actual access to #1.

AmbrosiaDB Globals

Handle necessary functionality for adminning through the admin-only globals and softcode.

The setup I personally suggest is to have ALL staffers be AT MAX Architect-level for everyday work and communication, with special coders that -really- require it having Councilor-characters available to log into for maintenance or special code setup. Technically if everything is done right, the Coder(s) of the game do not require higher privileges than Architect for the vast majority of things. Royalty-level code should be a rare exception, if at all necessary. The MU* Head(s) or site-admin should be the only one having access to #1, Immortals or perhaps even Royalty. The BC's, Global Function objects and Function objects at level Royalty and higher have simply been provided as a if-necessary convenience.

Current objects are only @chowned to certain bitlevels if it is really required for them to function. Whenever possible, they have been @chowned to the mortal BC-Admin-Mortal. All custom global functions listed above are on the semi- -mortal Global Functions object. The Master Room and Auxiliary Room have been @chowned to BC-Admin-Architect.

The Comsystem and +bboard are owned by BC-Admin-Architect, which means that higher bitlevels might not be able to use those systems if they hide and set themselves DARK. This is intentional. The Architect bitlevel is enough to freely set attributes on players, so these systems did not need anything higher, and it prioritizes Councilor+ as mere mainenance-duty bitlevels. Even the MU* Head should log on as an Architect for everyday things.

The Comsystem and BBOARD have been modified to be configurable by Architect and higher. Both systems have a CANUSE attribute with the according code on them. Note that if you want both systems to be only configurable by Councilor+, that instead of @chowning them to a Councilor after changing those attribute for Councilor or higher, I suggest to simply @set the bboard and comsystem core objects NO_MODIFY instead, keeping them at Architect-power but making them unmodifyable by Architects.

5.3.5 AmbrosiaDB Quota

I highly recommend the use of the alternative @quota system. BC-Admin-Mortal and BC-Admin-Guildmaster both have this @quota system set up on themselves. Both of them have a high amount of money for everyday operations. You should not give them free quota or money.

5.3.6 AmbrosiaDB Adding Functions

I also recommend to setup most global functions with /Privileged even if they are mortal-powered, to make them work even when players are set NO_CODE and WANDERER by default.

Enjoy!

5.3.7 AmbrosiaDB Compiling

P.S. the 'bin/asksource.save0' file has been supplied for loading in the 'make config' or 'make confsource' step of Rhost installation. It provides the settings I have used when creating this database. Some settings, like the ANSI substitution, are used in the DB.

–Ambrosia@RhostMUSH

CHAPTER

SIX

WHAT FLAGS, TOGGLES, POWERS, DEPOWERS, AND BITLEVELS MEAN IN RHOSTMUSH

6.1 What are Flags?

Flags are pretty much exactly the same as any other mush. It's a flag that you set or unset on a target which then enables/disables or alters something that target can do. There's help on all the flags in help and wizhelp.

6.2 What are Toggles?

Toggles were designed as a single point flag that immediately enables or disables a set ability or condition, thus a 'toggle'. It works exactly like a flag and was originally designed for two reasons. To distinguish from the multi-meaning of a 'flag', and because frankly we ran out of flag space :)

6.3 What is an @power?

A power is similar to a power on other mushes, but unlike them, our powers are multi-tier. This means that they can be customized to empower something at a given bitlevel. You may empower something from guildmaster up to councilor level. There are some powers with a power level of N/A meaning they are a toggle power granting an absolute power level as specified in the help for that power. This requires the INHERIT flag for non-players to inherit powers, however, a specific object can be granted a power as well.

6.4 What is an @depower?

This is the anti-thesis of @power. Also, depowers do not require inheritance. They also have priority over flags, toggles, and powers. You can use depower to remove or lower abilities and control from a target, even a full wizard (royalty) can be depowered.

6.5 Multi-tiered bitlevel systems

RhostMUSH offers a multi-tier bitlevel system. They go in order of presidence You do not have to use all these bits, only use what you want.

6.5.1 ghod (#1) <bitlevel 7>

This bitlevel can do everything. Only those who you trust with absolute power should have this. Period.

6.5.2 Immortal(i) <Level 6> - Basically #1

The only thing this bitlevel can not do is directly effect #1, set/unset some internal flags/attributes, and set/unset the immortal flag. These players can do EVERYTHING else. Treat this bit as you would treat #1. Only give it to those you know without a doubt you can trust.

- Can do everything except set some internal flags, effect #1, and set/remove the immortal flag.

6.5.3 Royalty(W) <Level 5> - FULL wizbit level

This is your standard wizard. They can do everything you're used to on other mushes that wizards can do. In addition, they also override all locks by default (this can be disabled), and they have an enhanced wizcloaking ability (which also can be disabled). They can also set all the lower bitlevels.

- All things of Level 4 and lower
- Ability to set more flags: STOP, NOSTOP, FUBAR
- Ability to @attribute,
- Ability to WIZCLOAK

6.5.4 Councilor(a) <Level 4> - High wizbit level

This is your almost-but-not-quite wizard. They have access to about 80% of the wizard commands. This includes @nuke, @toad, @newpassword and the like. The only things they can't do that wizards can is cloak, override locks, and use some of the database manipulation tools in wizhelp.

- All things of Level 3 and lower
- Ability to set more flags: NOCONNECT, WANDERER, FREE
- Ability to @nuke, @toad, @boot, @chownall, @dbck, @poor, @newpassword, @pcreate, slay

6.5.5 Architect(B) <Level 3> - Middle wizbit level

This is your sub-wizard. They still have the ability to control anything their bitlevel and lower (including @chown, @destroy, etc) but do not have any control of other players (like @nuke, @toad, etc) but they can set the slave flag. Otherwise, all things their level and lower they can treat as if they owned it.

- All things of Level 2
- Ability to fully control and modify anything their level and lower (including @cloning, @destroying, etc)
- Ability to use @tel on anything their level and lower.

- Ability to bypass jump_ok rooms on anything their level & lower.
- Ability to set some restricted flags: SLAVE, NO_YELL
- Has infinite quota and money
- Able to give negative money (Steal)
- Able to @toggle the MONITOR

6.5.6 Guildmaster(g) <Level 2> - Lowest wizbit level

This is the lowest wiz bit. They only have moderate abilities. They can examine/decompile anything their level and lower, they can @guild/@quota people, and they have a few other minor abilities. They don't have free money however.

- Ability to access things remotely (long_fingers)
- Things are FREE for them in the queue.
- Can see dbref#'s of things their level and lower
- Can examine/decompile things their level and lower.
- Can set @quota/@guild on their level and lower.

6.5.7 Wanderer <bitlevel 0>

This is a hinderance flag. This flag is automatically set on new players that are created (which can be disabled). This flag stops the player from creating/destroying any database information. In effect it stops them from any type of building type commands. They still are allowed to set/unset locks/attributes/etc though without hinderance.

6.5.8 Guest <bitlevel 0 as well>

This is a bigger hinderance flag. By default all guests should be set this. This flag stops the player from ANY database manipulation along with @teleporting, and many other advanced commands. It's extreamly dehibilatating.

6.6 Altering bitlevels

Please keep in mind each of these bitlevels can be tweeked with the @admin parameters and with the @powers (accessable by royalty) or @depowers (only by immortal and higher).

CHAPTER

SEVEN

SECURITY

7.1 Considerations to locking down restrictions in RhostMUSH

Sometimes, you want to have things run at various privilage levels and do not want to have things with too much access. Weither that is online objects or players you want to block from connecting to your mush. Here's things you can do.

One thing to keep in mind is that RhostMUSH, unlike PennMUSH is not flag dependant on permission level, it's ownership based. While setting a wizard flag on an object would work, it's not recommended and it is instead recommended to chown the object in question to a wizard (like your wizard holder character) Then the object must be set inherit to actually inherit the wizard.

Note: inherit is required to inherit anything from the player. Flags, powers, toggles. The only thing that is inherited automatically is depowers.

7.1.1 Online: Blocking object abilities

We have various flag levels. It is strongly recommanded you check wizhelp on 'control' for a detailed overview of what each bitlevel can or can not do prior to giving the ownership to the object. Things useful for tweaking control on players and objects:

FLAGS (access with @set)

IMMORTAL, ROYALTY, COUNCILOR, ARCHITECT, GUILDMASTER, FUBAR, SLAVE, SIDEFX, NO_CONNECT, WANDERER, SAFE, AUDITORIUM, BACKSTAGE, NOBACKSTAGE, INDESTRUCTIBLE, INHERIT, JUMP_OK, NO_TEL, NO_WALL, NO_EXAMINE, NO_MODIFY, NO_CONNECT, NO_POSSESS, NO_PESTER, NO_OVERRIDE, NO_USELOCK, NO_MOVE, NO_YELL, CLOAK, SCLOAK, DARK, UNFINDABLE, SEE_OEMIT, TELOK, SUSPECT, SPAMMONITOR

TOGGLES (@toggle)

BRANDY_MAIL, PENN_MAIL, MUXPAGE, VPAGE, NOISY, MONITOR* (all monitor toggles), MORTALREALITY, NODEFAULT, NO_FORMAT, PAGELOCK, SNUFFDARK, VARIABLE

@powers, @depowers, and @locks

Please review help (and wizhelp) for each of these items on how it can affect a player, thing, exit, or room. The help is quite verbose.

7.1.2 Offline: Blocking twinks from being abusive on your game

FLAGS : FUBAR, SLAVE, NO_CONNECT Commands: : @boot, @nuke, @toad, @turtle Sitelocks: (@admin) : forbid_host, forbid_site, register_host, register_site, noguest_host, noguest_site Sitelock by player : @badsite, @goodsite, NO_CONNECT (flag) Monitoring player : SUSPECT (flag), @snoop TOR/Proxy blocking: : @blacklist (see shell's tor_pull.sh), @admin proxy_checker (see wizhelp), @tor (see wizhelp)

7.2 Extended lockdown of the mush and considerations

These are flags, powers, toggles, and various conditions for consideration when you decide to use some of the advanced features of RhostMUSH. These are not all that is availble, but tend to be the juicier ones to consider.

7.2.1 Attribute Restriction

@attribute – Used for user-defined attributes @admin attr_access – used for built in attributes (like desc) @aflags – Used to set up lovely delicious attribute permission masks

7.2.2 Command Restriction

@icmd - Very useful. Please see wizhelp on it. It disallows commands from executing including overriding them with softcode alternatives @admin access - Changes permissions, disables, or sets to be overridden a command. Useful when you plan to override commands with softcode.

7.2.3 Flag/Toggle Restriction

@flagdef - Again, see wizhelp on this. There are also netrhost.conf options so you can have them loaded at start. This allows tweaking flags and toggles to who can set/unset/see as well as what type can use it or wha type it can be used on.

7.2.4 Config restrictions

@admin config_access - Changes permission of who can set a config param

7.2.5 Function Restrictions

@function/@lfunction – Allows softcoded functions that you can optionally lock down at your leasure
@admin function_access – You can use this even on softcoded functions if you so desired.

7.2.6 RhostMUSH Flags and Descriptions

```
GUEST       - This is your guest flag, it should only be set on guests
WANDERER    - the WANDERER flag is default on new players.  This flag disables all␣
↪building abilitites of the player.
NO_COMMAND  - You can use this to stop a player from being able to connect without␣
↪worrying about changing their password
FUBAR       - As the flag states, it f*'s them up beind all recognition.  It␣
↪essentially stops them from doing absolutely anything in the mush but pose and say.␣
↪ Yes, it even disables the quit command.
SLAVE       - Funny enough, slave allows anything but say and pose.  To ruin a troll
↪'s life, set both FUBAR and SLAVE and sit back and smile.
NO_TEL      - The target can't teleport or be teleported
NO_MOVE     - The target is locked at their location unable to move at all
NO_WALL     - They do not see any @wall except a wizard @wall/no_prefix.  This has␣
↪the bonus of snuffing db save messages.
NO_POSSESS  - Sometimes it's useful to grant a builder character to multiple players.␣
↪ The NO_POSSESS flag makes it so that player can not be logged in more than 2 times.
NO_MODIFY   - The target can not be modified (except by immortal/#1)
NO_EXAMINE  - The target can not be examined/decompiled (except by immortal/#1)
STOP        - Once a matching $command is found on an object set STOP, it 'stops'␣
↪trying to find other $command matches.
NOSTOP      - If a target that is set STOP is also set NOSTOP, it will check the␣
↪master room for a command and execute that as well if found.
NO_PESTER   - Stops target from @pemit or whisper.  You may use @icmd as well.
NO_OVERRIDE - Useful for immortals.  By default they override all locks, including␣
↪attribute locks.  This makes it so an immortal's passing of locks will behave like␣
↪a mortals
NO_USELOCK  - This is like NO_OVERRIDE but only effects uselocks.  You likely want to␣
↪set this on your immortal and wizard.
NO_ANSINAME - stops a target from having an ansified name
NO_CODE     - lock down advanced coding from a target
SPAMMONITOR - stop a target from issuing more than 60 commands a minute.
FREE        - Stop costing money for day to day processing of commands/building
```

7.2.7 RhostMUSH Toggles and Descriptions

```
MONITOR             - Enables site monitoring.  This is the main toggle
MONITOR_SITE        - Adds site information to site monitoring
MONITOR_USERID      - Adds the userid to site monitoring
MONITOR_STATS       - Adds connection stats to site monitoring
MONITOR_FAIL        - Adds showing failed connections to site monitoring
MONITOR_CONN        - Adds connection monitoring to site monitoring
MONITOR_DISREASON   - Adds disconnect reasons to site monitoring
MONITOR_TIME        - Adds time stamps to site monitoring
MONITOR_BAD         - Shows all bad creation attempts to site monitoring
MONITOR_VLIMIT      - Shows attempts to bypass MAX ATTRIBUTES
MONITOR_AREG        - Shows all auto registration attempts
MONITOR_CPU         - Shows all CPU warnings and/or alerts on the mush
```

(continues on next page)

(continued from previous page)

```
NO_FORMAT           - Bypasses @conformat, @exitformat, and other formats
SEE_SUSPECT         - Allows you to see suspect info in the WHO/DOING
FORCEHALTED         - Allows you to @force/@sudo a HALTED target
NOSHPROG            - Disallows using '|' to execute commands outside @program
PROG                - Allows the target to use @program
IMMPROG             - Disables the ability to use @quitprogram from a @program
PROG_ON_CONNECT     - Allows a @program to resume if someone reconnects
IGNOREZONE          - Enables a zone to process @icmd's
PAGELOCK            - Enforces target to require passing pagelocks
MAIL_LOCKDOWN       - Blocks the ability of a wizard to check another player's mail
ATRUSE              - Enables the attribute to use attribute content locking
NOGLOBPARENT        - Disables the target from inheriting global parenting
LOGROOM             - Enables system logs on the room
EXFULLWIZATTR       - Allows target to examine all wizard attributes
NODEFAULT           - Disables attribute formatting/handling on the target
CHKREALITY          - Enables the use of reality locks on the target
HIDEIDLE            - Disables deidling when you execute any command
MORTALREALITY       - Enforces a wizard to pass realities as a mortal
SNUFFDARK           - Hides dark exits from a wizard
```

7.2.8 RhostMUSH @powers and Descriptions

```
WIZ_WHO             - Allows target to see sites ala wizard who
NOFORCE             - target an not be forced (except by immortal/#1)
FREE_QUOTA          - Allow target to have unlimited quota
JOIN_PLAYER         - Allow to 'join' a player's location
NO_BOOT             - Player can not be booted except by immortal/#1
STEAL               - Player can give negative money
TEL_ANYWHERE        - Player can teleport anywhere
STAT_ANY            - Player can @search, @stat, or @find things
HALT_QUEUE_ALL      - Player can halt the queue
SEARCH_ANY          - Player can search for anything
WHO_UNFIND          - Player can see hidden player on WHO
SHUTDOWN            - Player can @shutdown the mush
PURGE               - Player can use /purge to @destroy and @nuke
EXAMINE_FULL        - Player can examine anything (not set NO_EXAMINE or cloaked)
FORMATTING          - @*formats allow passing what a person sees as %0, %1, etc
CHOWN_ANYWHERE      - Chown anything anywhere to yourself
CHOWN_OTHER         - Chown something you don't own to yourself
EXAMINE_ALL         - Examine other things (tiered)
SEE_QUEUE_ALL       - Player can see the full queue
GRAB_PLAYER         - Player can grab a remote player and pull them to location
LONG_FINGERS        - Player is granted remote control of things they own
BOOT                - Player can @boot
SEE_QUEUE           - Player can see advanced queue features
TEL_ANYTHING        - Player can @teleport anything
PCREATE             - Player can @pcreate players
NOWHO               - Allows the use of @hide
HALT_QUEUE          - Allows halting queue by specified bitlevel
SECURITY            - Allows setting  NOMOVE    NO_TEL   SLAVE   NO_YELL
WRAITH              - Allows bypassing exit locks
HIDEBIT             - Hides your admin level from lower levels
FULLTEL             - Allows full immortal level teleportation
EXECSCRIPT          - Allows executing external scripts in ~/game/scripts
```

7.2.9 RhostMUSH @depowers and Descriptions

```
WALL                - Disables the ability to @wall
LONG_FINGERS        - Disables remote access to things
STEAL               - Can not steal money
CREATE              - Can not create anything
WIZ_WHO             - Can not access wizard who
CLOAK               - Can not wizard cloak
BOOT                - Can not boot
PAGE                - Can not page
FORCE               - Can not @force/@sudo
LOCKS               - Can not pass locks
COMMAND             - Can not execute any $command (including master room)
MASTER              - Can not use any master room $command
EXAMINE             - Lowers or disables the ability to examine/decompile
NUKE                - Can not nuke, toad, or turtle
FREE                - No longer has free money for anything
OVERRIDE            - No longer can override anything
TEL_ANYWHERE        - No longer can teleport anywhere
TEL_ANYTHING        - No longer can teleport anything other than themselves
POWER               - Can no longer use @power
MODIFY              - Can not modify things
CHOWN_ME            - Can not chown anything to themselves
CHOWN_OTHER         - Can not chown anything to others
ABUSE               - Can not use $commands on anything they do not own
UNL_QUOTA           - No longer has infinite quota
SEARCH_ANY          - Disables the ability to @search/@find things
GIVE                - Disables ability to give things/money
RECEIVE             - Disables the ability to recieve things/money
NOGOLD              - Limits (or disables) how much gold someone can give
NOSTEAL             - Can not give negative gold
PASSWORD            - Can not change password
MORTAL_EXAMINE      - drops examine and all fetching to mortal only
PERSONAL_COMMAND    - Disables all $commands on anything they own
```

7.2.10 Site Restrictions

These are accessable via the @admin command, and the following options are allowable.

You may see all site information at any time with: @list sites

IP based restrictions

You may use CIDR notation such as /32 instead of 255.255.255.255. Config file: (see section on forbid_site as it describes and gives examples) Online Syntax: MASK:

```
                @admin forbid_site=123.123.123.0 255.255.255.0
                @admin forbid_site=123.123.123.123 255.255.255.255

CIDR::

                @admin forbid_site=123.123.123.0 /24
                @admin forbid_site=123.123.123.123 /32

REMOVING: MASK::
```

(continues on next page)

(continued from previous page)

```
                @site/all 123.123.123.123=255.255.255.255
                @site/forbid 123.123.123.0=254.255.255.0

          CIDR::

                @site/all 123.123.123.123=/32
                @site/forbid 123.123.123.0=/24
```

```
forbid_site      - Set the specified site forbid only
register_site    - Set the specified site register only
noguest_site     - Set the specified site unable to connect to guests
suspect_site     - Set the specified site suspect on connect
noautoreg_site   - Set the specified site to not allow autoregistration
trust_site       - Allow site to bypass suspect site restrictions
permit_site      - Allow site to bypass sitelock restrictions
nodns_site       - Site will no longer do reverse DNS lookups
noauth_site      - Site will no longer do AUTH ident lookups
```

DNS based restrictions

These allow globbing wildcard matches. The advanced feature is you can specify filtering on when the condition is matched, such as allowing 2 players from a site to be connected before disallowing anyone else to connect. Config File: (see section on forbid_host as it describes and gives examples) Online Syntax:

```
ADD:      @admin forbid_host=*.dsl*.comcast.net *.aol.com *another.site
DEL:      @admin forbid_host=!*.aol.com
ADVANCED: @admin forbid_host=mudconnect.com|3 (allow 3 at once only)
```

```
forbid_host      - Set the specified site(s) forbid only
register_host    - Set the specified site(s) register only
noguest_host     - Set the specified site(s) unable to connect to guests
suspect_host     - Set the specified site(s) suspect on connect
noautoreg_host   - Set the specified site(s) to not allow autoregistration
validate_host    - Do not allow any autoregistration from emails matching site
goodmail_host    - Always allow autoregistration from emails matching site
nobroadcast_host - Snuff online site broadcasts via MONITOR for specified site
```

7.3 Methods to block anonymous connections and the pros and cons of doing so

Now let's assume you have some troll attempting to use proxies to connect. There's multiple ways to stop this.

7.3.1 Blacklisting through external tor_pull.sh script

In ~/Rhost/Server/game you will see a script called tor_pull.sh Execute this by running (from the game directory) ./tor_pull.sh This populates the blacklist with registered proxies from various sites on the internet. If you want specified ip's added, feel free to add them at the end of this file.

7.3.2 Blacklisting through internal @blacklist command

On the mush, have as part of your startup @blacklist/load This will load in the generated blacklist file for automatic forbid sites based on the ip.

7.3.3 Blacklisting through internal @tor command

@tor. Please see 'wizhelp tor' on how to set this up. It in effect will actively block all known exit nodes for TOR's annonymous proxy service. It self-updates and will actively block TOR connections.

7.3.4 Blacklisting through internal @admin command

@admin proxy_checker (please see wizhelp) This little doodad uses MTU checking against packet size which will detect most methods of proxies. Sadly, this also has false positives because some situations require a differentating MTU value such as briged network connect with things like cloud services, docker, or similar encapsulated network services. However, this option has several settings from just monitoring/alerting of possible proxies to downright forbidding them. If you're being actively attacked, it may be worth considering adding this to add additional protection.

7.4 Setting up an SSL tunnel for secure connection options

7.4.1 Quickstart for SSL setup with stunnel

1. Modify your netrhost.conf file and add/change the following parameters:
    1. sconnect_reip 1
    2. sconnect_cmd SECRET-MAGIC-COOKIE
        1. SECRET-MAGIC-COOKIE is a case sensitive single word phrase. Any printable character other than the '#' character is allowable. You may use up to 30 characters.
        2. Make sure the secret is a hard to guess phrase. This is used by stunnel to forward on the originating IP address.
    3. sconnect_host localhost 127.0.0.1 othersite.goes.here
        1. This is optional.
        2. If you do not specify it it defaults to 'localhost 127.0.0.1'. If your domain has a unique name like 'localhost.localdomain' like some ubuntu distributions, then you should customize your stunnel_host.
2. go into the stunnel directory
3. ./stunnel_setup.sh
    1. Choose the defaults or alter them based on preferences
    2. Make sure to choose the warpbubble conf file

4. ./stunnel_start.sh
5. Use ./stunnel_stop.sh to stop the SSL tunnel at any time

You do not need to shutdown the ssl handler if you shutdown the mush. They are entirely separate processes.

7.4.2 Detailed SSL setup with stunnel

To setup SSL connectivity, we use the STUNNEL application to tunnel SSL to the mush. This acts a bit like a man in the middle but remains controlled by the game owner which would have access to the end point anyway.

Note: it is assumed you will have already initially set up your netrhost.conf.

---

stunnel directory

You would set up the stunnel from the 'stunnel' directory. There the following files are of relevance:

```
README                    -- a readme explaining the points of stunnel
stunnel.conf.example      -- The example stunnel.conf file.  If you wish to create␣
↪this manually you're welcome to.  Just make sure the end file is stunnel.conf
stunnel_setup.sh          -- the script to build a stunnel.conf file for you which␣
↪will be dropped at your specified location.
stunnel_kill.sh           -- Stop/terminate the stunnel process.
stunnel_start.sh          -- Start the stunnel process.
warpbubble.pl             -- the perl script that handles stunnel to mush␣
↪connectivity.
stunnel_src               -- If you do not have stunnel, this directory will allow␣
↪you to download, compile and locally install.
```

Modifying netrhost.conf

To be able to utilize SSL, you first must set your netrhost.conf file with the relevant information to enable SSL connectiions. These three config options must be set to be able to use SSL, however, sconnect_host if not set will default to 'localhost 127.0.0.1'.

```
sconnect_reip 1          -- This enables the SSL tunnel layer handler within rhost.
sconnect_cmd XYZZY       -- this will set the secret SSL command handshake command to␣
↪XYZZY.  This is case sensitive and can be up to 31 characters.  Please make sure to␣
↪only use printable non-whitespace characters.  Ergo: one word
sconnect_host wildcards -- This allows wildcarded sites (one or more) to allow to␣
↪access the sconnect/stunnel handler.  This defaults to 'localhost' and '127.0.0.1'␣
↪so if you have 'localhost.localdomain' instead then you must set this to whatever␣
↪is seen as 'localhost' to you.  You can verify this by checking your /etc/hosts␣
↪file.
```

Note: the sconnect_host is optional. If you do not specify it, it will default to two values: 'localhost' and '127.0.0.1'.

Running the stunnel setup program

At this point you're ready to run the stunnel setup program. So at this point type the following:

```
./stunnel_setup.sh
```

This will prompt you through settings. Most you can select the defaults to. The SSL port you may need to change based on your administrative requirements. It will prompt you with whatever you set for your mush name. If you have not selected a mush name at this point, you can select the defaults.

You will want to use the config file for warpbubble as this hides the secret.

Be aware that if you have DNS host lookups disabled on your mush, you MUST have 127.0.0.1 as an entry for your sconnect_host file.

Starting the stunnel proxy

When you have your stunnel.conf file to the way you want, you then issue the following command to run your SSL layer:

```
./stunnel_start.sh
```

Shutting down the stunnel proxy

If ever you need to bring down the SSL layer, you may kill it with the command:

```
./stunnel_stop.sh
```

Configuring firewall on the host

Please be aware that the port that the SSL layer is on must be opened in any firewall rule you specified to allow the connectivity. This also must not be the port the mush is running on and requires a separate port.

CHAPTER

EIGHT

MAINTENANCE

8.1 Note about Patching

There's two ways you can look to patch the source. If you plan to run the RhostMUSH source from a git repository, then please use the git repo to constantly update your code. If you knew enough to want to set up a git repo then we expect knowledge on how to keep source trees updated in the git repo to be used the same as any other source distribution.

If, however, you have no idea what a git repo even is, or have no inclination of using git to manage your RhostMUSH source, or just don't care one way or another, then you can use the included patch.sh routine (from under the Server directory) to patch your source at any time.

From the server directory just type:

```
./patch.sh
```

That will auto-compile your source, auto make all your header files and essentially keep everything up to date to the latest source. Once that's done, all you do from within the game is two commands:

```
@reboot (or @reboot/silent)  -- This will load in the new binary
@readcache  -- This will read in all the .txt file changes
```

8.2 Daily Backups for RhostMUSH

Make SURE YOU RUN DAILY Backups. Rhost is very stable, but things outside the mush can damage the game. paranoia is fine, especially when they really are out to get you. TO make the backups, do the following:

```
 @dump/flat      -- This makes a flatfile dump of the main database.  You want to run␣
↪this daily.
 wmail/unload    -- This makes a flatfile dump of the mail database.  You want to run␣
↪this daily.
 @areg/unload    -- Only worry about this if you are using auto-registration emailing.
↪  Few do.
 newsdb/unload   -- Only worry if you use the hardcoded bbs system.  Most don't use␣
↪it.


Backups are already handled and integrated with a script 'backup_flat.sh'.
If you wish to customize this, feel free.  Again, it is well documented and
just require changing settings at the top of this script.
```

(continues on next page)

(continued from previous page)

```
By default, it does 7 contiguous backups.  You may increase or decrease
this value to any value you want.

It will, by default, backup all your txt/\*.txt files, your netrhost.conf
file, your netrhost.db.flat (mush db) file, your RhostMUSH.dump.\*
(mail db) files, your RhostMUSH.news.\* (internal news/bbs db -- if used),
your RhostMUSH.areg.\* (the autoregistration db -- if used), and any sqlite
database you currently may be using which are OPTIONALLY backed up if you
remove the '#' from before it.

The backup script also will optionally rcp/scp, ftp, or mail any backups
you want to a remote destination.  Be forewarned, the backup files can
potentially get rather large for larger games, even compressed.  The
average size for these files will be 1-5MB.  It could potentially get
over 10-20MB in size for excessively large games, so plan accordingly.

Be aware that the backup system will NOT make successful backups if you
run out of disk space.  This includes actually running out of disk space
or running out of disk quota.  There is a mechanism inside the backup
script to specify an email address that you wish to get alerts from
in these instances.  I recommend using it.

If you make changes to your backup_flat.sh script with an already
active and running mush and wish to just restart the backup procedure
just issue::

   ./backup_restart.sh
```

8.3 Signals and why you need them for control

Rhost by default allows signals at the shell to handle various processes in-game.

The following signals are useful.

8.3.1 TERM (kill -TERM or kill -15)

- This will immediately terminate the mush, dumping a special flatfile called netrhost.db.TERM and scramming the db in question by force-closing it without any writes. A TERM is the effort for the mush to shut down the mush as fast as possible to avoid any db corruption if possible since a TERM signal is common during a server shutdown, so time is paramount.

8.3.2 USR1 (kill -USR1)

- This by default issues a reboot on the server. This is a special parameter because this can actually be changed in-game to do any number of other things. Please refer to the RhostMUSH running in question if this is the default behavior or if the method for USR1 is doing something else.

8.3.3 USR2 (kill -USR2)

- This will shutdown (cleanly) the mush. This behaves as if you issued a @shutdown from within the game, and follows all proper procedure in bringing the game down cleanly and safely. This shoudl be used when doing maintenance on the game or if you need to bring it down from the shell.

8.3.4 KILL (kill -KILL or kill -9)

- This signal can not be caught and will immediately terminate the game without any safty to the database at all. Short of something horribly wrong going on, this should never be used to bring down your mush. Doing so will almost certainly corrupt your databases (ALL OF THEM) that are open, including but not limited to your main database, your mail database, your autoregistration database, and so forth. So if you do this, plan to do some database recovery from your flat files. Also, when you bring down a mush in this manner, you need to issue Startmush -f to bring it back up.

8.4 Shutting down RhostMUSH gracefully

8.4.1 RhostMUSH has many ways to shutdown the game cleanly

1. Log into the mush and issue @shutdown
2. Issue a kill -USR2 to the mush which issues an emergency @shutdown
3. Issue a kill -TERM to the mush which issues an emergency abort and clean shutdown.

8.4.2 WARNING: Never kill -9 RhostMUSH

Under NO CIRCUMSTANCES should you kill -9 your mush unless you don't care for the database. The reason is if the mush happens to be saving, in any method, to the database, especially a QDBM database, you will likely have just corrupted your database, so pull out a flatfile to recover.

Sadly, this also may occur if the server hosting you takes a nose-dive in the middle of a db write. Rhost can recover corruption in-game while up, but if it bombs in the middle of a write, all bets are off. :)

8.5 Autoshutdown script

The makefile will 'make' the program that will STOP the mush. Please tweek 'autolog.h' with the proper parameters.

The 'startup.sh' will START the mush.

Both of these are intended to be used for automations (automated processes) like your rc.local file and/or startup scripts when you bring your server up.

8.6 Network Port redirector

This is a port redirector incase you decide to move your mush to a new site/port. To use, first, compile the code. To do this you would type the following:

```
cc portmsg.c -o portmsg
```

if 'cc' is not defined, try the following:

```
gcc portmsg.c -o portmsg
```

Once compiled, you would then modify the file 'file' to describe the mush, what was done, where it's moved to, then specify the IP address and the PORT where specified.

To launch the application, you would then type:

```
./portmsg file <port>
```

Where <port> is the port where the mush used to run at.

8.7 Using the built-in cron system for periodically running commands

8.7.1 Syntax for rhost.cron

The rhost.cron file will be in the syntax as follows:

```
username (or dbref#)
command1;command2;command3;...;commandN
command
command
command1;command2;command3;...;commandN
```

You can have commands strung together with a semicolon on the same line. This counts as a single line of input. You can have at most 20 lines of commands after the target you wish to execute the commands as. The target may be a player name OR a dbref# of any valid dbref# within the game. Invalid targets will abort the cron process. Non-printable characters in the cron file will abort the process. Any aborts or warnings will be logged in the netrhost.gamelog.

Here is a working example of the code cron file. This example will perform dumps of the mush.

Example syntaxt for rhost.cron

```
#1
@dump/flat; @@ dump the main game database to flatfile
wmail/unload; @@ dump the mail database to flatfile
@areg/unload; @@ dump the registration database to flatfile
newsdb/unload; @@ dump the news bbs database to flatfile
```

8.8 The following scripts are used in the game directory

Startmush – used to Start up the mush backup_flat.sh – Used to run backups with @dump/flat within the game (Started with Startmush automatically) backup_restart.sh – Restart the backup_flat.sh if changes are done compress_logs.sh – Compress logs in 'oldlogs'. Ran with Startmush findit.sh – Internal script used to check for flatfile validity mailhide.sh – Wrapper to hide from address using the 'mail' progam minimal.sh – Auto-load the minimal db into the mush proxysnarf.sha – Internal script for the tor_pull.sh tor proxy blacklist tor_pullit.sh – Internal script for the tor_pull.sh for proxy blacklist recovery.sh – If your db is corrupt, run this to auto-revert to an earlier flatfile tor_pull.sh – Create a blacklist.txt file that can be loaded via the internal @blacklist command for proxy handling

8.9 Textfiles for RhostMUSH

areghost.txt - file player gets when autoregistration on registered host. autoreg.txt - file player gets when autoregistration on non-registered host. autoreg_include.txt - file player receives in email when they autoregister attached to login/passwd badsite.txt - file player gets when site is not allowed. connect.txt - file player gets when connect create_reg.txt - file player gets when their site is register and they can't create. doorconf.txt - file that is searched for information regarding @door.

Note: Need to mkindx doorconf.txt doorconf.indx for this file

---

down.txt - file player gets when the mush has logins disabled (@disable login) error.txt - the 'Huh? (type help for help)' messages.

Note: Need to mkindx error.txt error.indx for this file

---

full.txt - file player gets when the mush can't have any more players. guest.txt - file player gets when they connect as a guest. help.txt - your help file

---

Note: Need to mkindx help.txt help.indx for this file

---

motd.txt - your motd file news.txt - your news file

Note: Need to mkindx news.txt news.indx for this file

---

newuser.txt - file newly created players get when they connect for the first time. noguest.txt - file player gets when they are not allowed to connect to a guest. plushelp.txt - optional +help file. (needs compile time option)

Note: Need to mkindx plushelp.txt plushelp.indx for this file

---

quit.txt - file player gets when they disconnect. register.txt - file player gets when the site is locked down for registration. wizhelp.txt - your wizhelp file

---

Note: Need to mkindx wizhelp.txt wizhelp.indx for this file

---

wizmotd.txt - your wiz motd file

8.9.1 Textfile Frequently Asked Questions

Q: How do I put color in these files? A1: Look at ansi.h and you need to put the literal ASCII codes. They will look like: ^[[0m (for ANSI_NORMAL). That's <ESC>[A2: You can enable ansi_txtfiles then use %c (or %x/%m) encoding for ansi, however you compiled your Rhost.

Q: I want to design my own txt files to read in the mush. A: Easy. Design them like help.txt would be set up, mkindx the file, then you can access it via @dynhelp online.

Q: Do I have to mkindx these files whenever I make changes? A: Only the ones that have '& ' index. (help.txt, wizhelp.txt, news.txt, etc)

Q: Do I have to @readcache in the game whenever I make a change? A: Only when you modify any of the files listed in README.TXTFILES. Not the ones you use with @dynhelp.

Q: Can't I just make code in the mush that then is used for these silly txt files? A: Absolutely. Check 'wizhelp file_object'.

CHAPTER

NINE

TROUBLESHOOTING

9.1 Reporting bugs or getting help

If you find any bugs or problems, notify one of the developers of RhostMUSH and a patch or workaround will be made available as soon as possible. Current developers are: Seawolf, Thorin, Ashen-Shugar, Lensman, Kale, Mac, Zenty, Ambrosia, Amos, and Morgan. They can be found around the net.

9.1.1 Troubleshooting issues with starting up

Problem: If it says the shared ID is already in use

A1: please verify that it is the right shared debug_id in your netrhost.conf file

A2: Force a start by running:

```
./Startmush -f
```

Problem: Your log file is massive and your mush is running

A1: To rotate this use the @logrotate command. See wizhelp on @logrotate

Problem: The database flatfile you're loading can't load because a db is already defined

A1: remove netrhost.db* and netrhost.gdbm* from your data directory

Problem: The mail database won't load and mail shows 'offline'

A1: from within the MUSH run:

```
wmail/load
```

9.2 Stack limit and debugging

Rhost uses a stack limit in the debug monitor.

This stack limit is set to a reasonable amount of 1000. This is defined in the debug.h file in the hdrs directory.

This will directly impact the function_recursion_limit from being set above 100. If, for whatever reason, you really must have a ridiculously high recursion limit, then it is a suggestion to manually modify the stack limit in debug.h to a higher number.

We have reasonably set it to 10000 without too much issue, but keep in mind, the overhead is higher for every stack you throw on the process table. Higher stack means more memory used.

Also be aware that your shell stack limit directly is affected to this value.

Type:

```
ulimit -a
```

This will show you your shell stack limits. Do NOT set the STACKMAX value higher than your shell's stack value.

The value in ~/Rhost/Server/hdrs/debug.h is currently set as:

```
#define STACKMAX 1000
```

Feel free to change this to a higher value if you wish.

The caveat. This effects the debug stack daemon. Meaning, the only way for this to be updated is through @shutdown and then a fresh ./Startmush.

A @reboot WILL NOT LOAD IN A NEW DEBUG MONITOR!!!!

You can issue @list stack to see the current stack ceiling ingame.

9.3 How to reset the password for #1

Warning: You can only use one of these options at a time. Make sure to change back your nerhost.conf and then reboot after making the changes.

9.3.1 Method 1: Reset to Default Password

in your netrhost.conf file add:

```
newpass_god 777
```

This will reset #1's password to the default 'Nyctasia'.

9.3.2 Method 2: Increase Permissions of Immortals

in your netrhost.conf file add:

```
newpass_god 1
```

This will allow IMMORTAL players to @newpassword #1 upon reboot.

9.4 Troubleshooting difficulties compiling RhostMUSH

9.4.1 Changes to conf for high-bit CPUs

RhostMUSH automatically detects 64-bit platforms, and should compile cleanly on these.

In case you are trying to compile Rhost on some other crazy-wide CPUs such as the PS2, PS3 or other 128 or 256 bit CPUs, you can easily do so by changing a few lines of code in conf.c.

change:

```
typedef unsigned int    pmath1;
typedef int             pmath2;
#define ALLIGN1 4
```

to:

```
typedef unsigned long   pmath1;
typedef long            pmath2;
#define ALLIGN1 8
```

Note: Replace 8 with the size of your CPU's long integer. (4 for 32-bit, 8 for 64-bit, 16 for 128-bit, etc etc)

RhostMUSH has only been tested to work on the AMD64, but there is no reason to believe the same will not hold true for IA64.

9.4.2 Changes to autconf for certain systems

You should not have to worry about this, but incase something really weird occurs, you may need to look into these changes...

The autoconfig.h file needs to have modifications to it by hand.

There are three manual entries:

This one sets how it defines the int to character pointer. It's safe to keep it as 'unsigned int' for 32 bit platforms. For non 32-bit, define it to how an int is defined on that system:

```
typedef unsigned int    pmath1;
```

This one sets how it defines the signed int to character pointer. Same restrictions apply as above for unsigned int:

```
typedef int     pmath2;
```

This sets the allignment for the given platform. 4 represents a 32 bit platform. 8 would represent a 64 bit platform, etc. Change accordingly:

```
#define ALLIGN1 4
```

Warning: Make sure these three entries are defined in your autoconf.h file else the mush will not compile.

9.5 Dealing with DB Corruption

Ok. Your database won't come up.

If you are reading this, then likely the scenerio is one of the following:

1. The mush says it can't find your database files.
2. The mush says it can't read or load your database files.
3. The mush seems to load fine but I can't log in anyone and when I do all the names and attributes of things seem to be gone!
4. Bringing up your mail database

First thing is first. Don't have a panic attack.

9.5.1 If the mush says it can't find your database files

Check the names of the database files in your 'data' directory

They should be named something like:

```
netrhost.db
netrhost.db.old
netrhost.db.old.prev
netrhost.gdbm.dir
netrhost.gdbm.pag
```

And you may see a netrhost.db.flat

Check your netrhost.conf file

If you never touched the *database or muddb_name params, you should be good.

Verify your *database params (and muddb_name) are still set to 'netrhost' as part of the name. Ergo, the default values and you didn't change them. These should match up with the filenames in your data directory.

If these names do not match up, it can't find the database files to load. So you shouldn't have to change these names, ever. :)

Check your mush.config file

If you never modified this file, you should be good.

The gamename should be 'netrhost' for this file. This does NOT control the name of your game. This controls the name of all the files as associated to the mush. So changing this means the netrhost.conf file, all your database files, and so forth. Please don't change this :)

9.5.2 If the mush says it can't read or load your database files

Double check everything for the previous issue. Make sure everything is named properly.

Verify you have enough disk space. (quota)

Some account have a limited quota to run in. If you reached or exceed your disk quota, you can have a corrupted database. So always keep your eye on the size. quota -s to see a human readable format to see how much quota you have left. You want to make sure current in use is below the 'grace' and soft/hard limits shown. If not, you're out of space.

You will need to remove some files before you repair and bring up your mush again. Try to keep your quota at least 200 megs free to allow plenty of wonderful growth space for the mush.

Verify you have enough disk space. (system)

The second way you can run out of disk space is by the filesystem itself. do a df -h . in your 'game' directory'. That is df -h <period>. This will return how much disk space is being used and how much remains. If it shows 100% used, you're out of disk space and the db is corrupt.

At this point, you're pretty screwed. You can see if anything exists in your system to free up some space, but if the filesystem itself is filled, reach out to the owner of the server and let them know. It's a much bigger deal than just your mush if that's the case.

Until this issue is resolved, you can not repair and bring up your mush. No disk to run the game.

9.5.3 If the mush seems to load fine but I can't log in anyone and when I do all the names and attributes of things seem to be gone!

Ok, at this point you likely had your mush up when the physical server went down hard. Weither through an emergency shutdown or a physical power outage, your db likely was brought down hard during a write, so it left it in a corrupt state. These things happen. This is why we always strongly request you make daily flatfile dumps.

So, to recover your database.

The bad news

If you have no flatfile backup or never bothered with backups? I'm sorry, at this point you're SOA. There's no easy way to recover a corrupted binary database. If you absolutely need data out of it we may be able to help you to piece meal things out of it, but otherwise it's a lost cause. You'll have to start over from scratch. I'm sorry.

The good news

If you made backups, or if the server had a normal shutdown, you likely have a flatfile backup. You will see a netrhost.db.flat in either the 'data' directory or 'prevflat' directory. That is your manual flatfile backup.

If the server had a normal shutdown, you will see a file called netrhost.db.TERMFLAT. This is a scram-emergency db flatfile. It attempts to write this at the time of server shutdown to hopefully keep a clean backup in the case of issues since it identifies the server is coming down hard. Make sure if you plan to use the TERMFLAT as your recovery flatfile that the very last line shows something like ** END OF DUMP **. That shows you had a successful backup.

Now, to restore your database?

Please refer to the file 'README.DBLOADING'.

9.5.4 Bringing up your mail database

Your mail db may or may not come up at this point.

If after restoring main database your mail database works

If your mail database came up and does not show 'Mail: mail is currently off' then you should be good to go.

Please issue on the MUSH:

```
wmail/fix
wmail/lfix
```

This will put your mail system in sync with your current database and fix up any errors that may exist.

wmail/fix fixes the mail.

wmail/lfix loads in the fixes.

If after restoring main database your mail database does not work

If your mail database is not up and shows 'Mail: mail is currently off' then your mail db is currupt.

9.6 Dealing with a corrupt mail database

It says when you try to access mail that mail is disabled and/or off.

Nothing you do can bring it on line. Well, this is bad, but not horrible.

The mail db is totally separate from the main game database. This means that it in no way damaged or corrupted your main mush database.

The bad news? Yes. Your mail database is corrupt. To bring it back, is it hopes that you read ahead of time about how to backup your mush, which would include the mail database.

9.6.1 Backing up your mail database

You should be making a flatifile dump of mail db daily by running on the MUSH:

```
wmail/unload
```

To recover your mail, it assumes you have a mail flatfile in either the ~/Server/game/data directory or the ~/Server/game/prevflat directory. The latter directory is used in junction to the backup_flat.sh and will always house the latest flatfile if not one recently dumped in your data directory.

9.6.2 Automatically recovering your mail database

If you have a flatfile dump of your mail db, run this command on the MUSH:

```
wmail/load
```

Yup, that's it. It'll take care of everything else. Isn't automation grand?

Doesn't even require a reboot :)

Note: You may at this point wish to run the following: wmail/fix – this fixes the mail database and sync's it to the mush db. wmail/lfix – this loads in the fixed mail database

---

If you have a very old mail database, this is likely going to be required to sync against nuked players and other changes to the game since the flatfile.

If this is a new db that you have, you can skip the fixing.

9.6.3 Manually recovering your mail database

To recover your mail manually, you need to delete your mail databases, reboot, then reload your mail flatfiles. If you have no mail flatfiles, well, you're going to have to start over with the mail database. Sorry.

First, go into the 'game' subdirectory. Inside that directory is a 'data' directory.

You will be deleting all the files with the following names:

```
RhostMUSH.mail.*                (like RhostMUSH.mail.dir/RhostMUSH.mail.pag)
RhostMUSH.folder.*              (like RhostMUSH.folder.dir/RhostMUSH.folder.pag)
```

Warning: DO NOT DELETE OTHER NAMED FILES!!!

Once these files are deleted, you may issue a @reboot to restart the mush. This will unlock the mail system and load in a fresh db.

Now, if you have flatfiles of the old mail database, you will see in either the 'data' directory or the 'prevflat' directory files that are called:

```
RhostMUSH.dump.folder
RhostMUSH.dump.mail
```

Make sure these two files are in the 'data' subdirectory. Copy them in if they exist in your 'prevflat' directory.

Once they are in the 'data' directory, within the mush type: wmail/load

This loads in the flatfile and recover the mail database.

Now, at this point the mail db may not be 100% in-sync with the game db.

So let's fix it.

wmail/fix – this will run a fix on the mail db and repair any issues.

wmail/lfix – this will load the fixed flatfile back into the mush.

At this point you should be good to go.

CHAPTER

TEN

ADVANCED FEATURES

10.1 Installing using an ansible playbook

To begin, you will run the following command in a directory that will house your game:

```
git clone https://github.com/RhostMUSH/trunk Rhost
```

You may also just run the yml file and ansible (ansible-playbook) to install your RhostMUSH engine:

```
wget https://raw.githubusercontent.com/RhostMUSH/trunk/master/rhostinstall.yml
ansible-playbook rhostinstall.yml
```

This downloads the latest stable version of the code, bringing with it all patches and scripts, documentation and support tools that you will need.

10.2 Adding hardcoded modules

RhostMUSH does support module writing.

10.2.1 Modifying sourcode to add a module

This requires hooking your changes into local.c, then modifying the Makefile (in the src directory) for any new source files that you wish to add.

Something to be aware of is that all localized data is ran after the system cache subroutine.

10.2.2 Adding an @startup to make use of modules

This means that if your code is depending on @startups, you need to put a delay in the @startup so that your local code can be loaded in as modules prior to the @startup execution.

Something that will not work:

```
@startup me=@superhappyfuncommand loadmeup=now
```

A small alteration that will likely make this work fine:

```
@startup me=@wait 1=@superhappyfuncommand loadmeup=now
```

That 1 second delay for the queue will give the game engine time to load in your module code.

10.2.3 Contributing your module back to Rhost

If you wish your modules to be part of the main Rhost distribution you have two options:

1. Attempt to hack the bin/asksource.sh and bin/asksource.blank files.
2. Ask one of the Rhost devs to do it for you :)

10.3 Reality Levels Setup

Reality levels are a means to forbid (or allow) interaction between objects in the same location.

10.3.1 Reality Levels Visibility

Each object (player, room, exit, thing) has two lists of reality levels. An Rx list, which describe what it can see and a Tx list, which describe where it can be seen. Those are bitfields. In order for X to see Y a bitwise 'and' is performed between X's RxLevel and Y's TxLevel. If the result is not 0, then X sees Y. If the result is 0, as far as X is concerned, Y doesn't exist. This affects contents lists, exit lists, look, say, pose, @emit, @verb, connect/disconnect, has arrived/has left messages, exit and object matching. 'here' and 'me' match always.

It doesn't affect @remit, @pemit, page, WHO or channels. By default, all new objects are created with an RxLevel of 1 and TxLevel of 1. Rooms are an exception, created with an RxLevel of 1 and a TxLevel of 0xFFFFFFFF. Those default levels can be changed with configuration parameters. An object is always visible to itself, even if its Rx and Tx levels don't match. (See examples below)

10.3.2 Reality Levels Descriptions

For every reality level defined, you can define an attribute that serves as description. If you look at something and match more than one of its TxLevels, you'll see all the corresponding descriptions on the target object. If the object doesn't have any descriptions for the matching levels, you'll see the regular @desc.

The @adesc attribute on the target is only triggered if the target can see the looker in turn. It's only triggered once, no matter how many descs the looker sees. The @odesc is shown only to those people that see /both/ the looker and the target.

Through extension, @afail/@ofail and similar pairs (@adrop/@odrop, @asucc/@osucc etc) work in the same way. @verb commands are similary affected.

Softcoded commands are only matched on the objects that can see the player. The player doesn't have to see the object. This includes commands in the Master Room.

Exits are treated specially. In order to be able to use an exit name (or to use the 'move <exit>' command) the exit must be visible to the enactor. In order to pass through the exit, the exit must see the enactor in turn. There are reasons for this, which will become evident in the examples below.

10.3.3 Reality Levels Configuration parameters

A few configuration parameters have been introduced to deal with the reality levels:

```
reality_level <name> <value> [<desc attribute name>]
```

This directive can only be used in the config file (not with the @admin command) and should be repeated for each reality level you want to define. It defines a new level named <name> with a bitvalue of <value> and an optional desc attribute. There is a limit of 8 characters on <name>, a 32-bit value on <value> (basically an unsigned long) and 32 characters on the attribute name. A maximum of 32 reality levels can be defined:

```
def_exit_tx <value>
def_exit_rx <value>
def_room_tx <value>
def_room_rx <value>
def_player_rx <value>
def_player_tx <value>
def_thing_rx <value>
def_thing_tx <value>
```

These 8 directives define the default reality levels of newly created objects. They can be set in the config file or with the @admin command. Like above, <value> must be a decimal number:

```
wiz_always_real <0|1>
```

If this parameter is set to 1 then wizards (and immortals on Rhost) will see everything and will be visible to everyone. Their effective Rx and Tx levels will always be 0xFFFFFFFF. Also settable in the config file and with the @admin command.

Compile with -DREALITY_LEVELS compile time option to enable 'Real' needs to be '1' This is an example file only to be added to the mush.conf file Format:

```
reality_level <8 char name> <hex-byte-mask> <optional-desc: DESC default>
```

10.3.4 Reality Levels Example mush.conf

```
reality_level Real 1
reality_level Obf1 2
reality_level Obf2 4
reality_level Obf3 8 OBFDESC
reality_level Obf4 16 OBFDESC
reality_level Obf5 32 OBFDESC
reality_level Obf6 64 OBFDESC
reality_level Obf7 128 OBFDESC
reality_level Obf8 256 OBFDESC
reality_level Obf9 512 OBFDESC
reality_level Obf10 1024 OBFDESC
reality_level Umbra 2048 UMBRADESC
reality_level Fae 4096 FAEDESC
reality_level Shadow 8192 SHADOWDESC
reality_level Spy 16384
reality_level Death 32768 DEATHDESC
reality_level All 4294967295
```

10.3.5 Reality Levels Commands

Two wiz-only commands are used to set the reality levels of an object:

```
@rxlevel <object>=<list>
@txlevel <object>=<list>
```

<list> is a space-separated list of level names that have to be set on the object. If a level name is prefixed with an exclamation mark (!) that level will be cleared from the object.

Warning: Changing the Tx levels of an object might make it invisible to you. In this case, you can still manipulate it by using his #dbref (or *player for players).

10.3.6 Functions

There are five functions that deal with reality levels:

```
hasrxlevel(<object>,<level>)
hastxlevel(<object>,<level>)
```

These two functions check if an object has the specified Rx or Tx level. You must control <object>. They return 0 or 1 and #-1 in case the object does not exist or you don't have permissions:

```
rxlevel(<object>)
txlevel(<object>)
```

These two functions return a space-separated list of the object's Rx or Tx levels. Again, you must control the object:

```
cansee(<obj1>,<obj2>)
```

A wiz-only function, returns 1 of <obj1> can see <obj2> from a reality levels point of view. It doesn't check if the objects are in the same location, the DARK/CLOAKED flags and so on. Just <obj1>'s Rx level against <obj2>'s Tx level.

Warning: If you are using it on MUX2.0 with /both/ reality levels and Wod Realms enabled, the function will perform both checks and the Wod Realms version checks against the DARK flag.

10.3.7 Example 1: A simplified Witchcraft setup

In Witchcraft, besides the various Gifted classes, characters can be spirits There are spirit realms to which the mundane can not travel. Therefore we will use 2 reality levels: Real and Ghost. Since some spirits can become solid for a limited period of time, we will also use an additional desc for the Ghost level, called GHOSTDESC. Therefore in the config file we will have:

```
reality_level Real 1
reality_level Ghost 2 GHOSTDESC
```

Ghosts can pass through most mundane locks, so the exists should allows the ghosts to pass:

```
def_exit_rx 3
```

Note that def_exit_tx isn't set. Why? Because ghosts see the mundane world anyway, so a spirit character will have:

```
@txlevel <player>=!Real Ghost
@rxlevel <player>=Real Ghost
```

Let's assume 3 players:

```
John is a Mundane. He won't see spirits.
John's Rx: Real
John's Tx: Real
John's @desc: This is John.
John's &GHOSTDESC: (Not important, since it's never visible)
Johh's @adesc: %N has looked at you.
John's @odesc: has looked at John.
```

Jack is a Gifted. He will sense spirits, but is still made from flesh and blood so visible to mundanes:

```
Jack's Rx: Real Ghost
Jack's Tx: Real
Jack's @desc: This is Jack.
Jack's &GHOSTDESC: (Not important, since it's never visible)
Jack's @adesc: %N has looked at you.
Jack's @odesc: has looked at Jack.
```

Frank is a ghost. He will see other spirits as well as mundanes, but won't be visible to mundanes. He can also become visible to everybody:

```
Frank's Rx: Real Ghost
 Frank's Tx: Ghost
 Frank's @desc: This is Frank, looking human.
 Frank's &GHOSTDESC: This is Frank's ghostly shape.
 Frank's @adesc: %N has looked at you.
 Frank's @odesc: has looked at Frank.
```

Following are commands that each of the players enter and what they see. I'll assume the +materialize command is defined like:

```
&CMD_MATERIALIZE <cmdobject>=$+materialize:@txlevel %#=Real; @pemit %#=You are now␣
↪material.
```

```
|          John            |          Jack            |          Frank
                           |                          |
 > l                       |                          |
 A room                    |                          |
 This is a bare room.      |                          |
 Contents:                 |                          |
 Jack                      |                          |
 Obvious exits:            |                          |
 Out <O>                   |                          |
                           |> l                       |
                           |A room                    |
                           |This is a bare room.      |
                           |Contents:                 |
                           |John Frank                |
                           |Obvious exits:            |
                           |Out <O>                   |
                           |                          |> l
```

(continues on next page)

(continued from previous page)

```
                          |                        |A room
                          |                        |This is a bare room.
                          |                        |Contents:
                          |                        |John Jack
                          |                        |Obvious exits:
                          |                        |Out <O>
>l Jack                   |                        |
Jack                      |John has looked at you. |John has looked at Jack.
This is Jack.             |                        |
>l Frank                  |                        |
I don't see that here.    |                        |
                          |>l Frank                |
                          |Frank                   |Jack has looked at you.
                          |This is Frank's ghostly |
                          |shape.                  |
                          |                        |>l John
                          |Frank has looked at     |John
                          |John.                   |This is John.
                          |                        |>+materialize
                          |                        |You are now material.
>l Frank                  |                        |
Frank                     |John has looked at      |Frank has looked at you.
This is Frank, looking    |Frank.                  |
human.                    |                        |
                          |>l Frank                |
Jack has looked at        |Frank                   |John has looked at you.
Frank.                    |This is Frank, looking  |
                          |human.                  |
                          |This is Frank's ghostly |
                          |shape.                  |
```

10.3.8 Example 2: A WoD setup

The reality levels will be defined like this:

```
reality_level          Real 1
reality_level          Obf1 2
reality_level          Obf2 4
reality_level          Obf3 8 OBFDESC
reality_level          Obf4 16 OBFDESC
reality_level          Obf5 32 OBFDESC
reality_level          Umbra 64 UMBRADESC
reality_level          Fae 128 FAEDESC
reality_level          Shadow 256 SHADOWDESC
reality_level          All 511
```

5 levels of Obfuscation, Umbra, Dreaming, Wraiths. 'All' is a handy replacement for all levels, useful for wizards and wizobjects that should be visible on all levels. Also useful when you want to set an object's levels to something without knowing what he had before:

```
@rxlevel #276=!All Real
```

!All will clear all levels, then the object will gain the Real level. There is more than one Obfuscation level because of the relation between Auspex and Obfuscation.

A vampire with Obfuscate 2, should not be visible by one with Auspex 1. However one with Auspex 3 should see another vampire with Obfuscate 1, 2 /or/ 3.

Obfuscated players can move if they have Obf > 1. Umbral and Shadow players should also be able to see most of the exits. So the exits at creation should have default levels of Real + Obf2 + Obf3 + Obf4 + Obf5 + Umbra + Shadow = 1 + 4 + 8 + 16 + 32 + 64 + 256 = 381:

```
def_exit_rx 381
def_exit_tx 381
```

Obf1 is not included since an Obfuscated vampire should not be able to move if it only has Obf1. Therefore they won't see the exits. If you want them to be able to see the exits, but not to use them, change the default Tx of the exits:

```
def_exit_rx 381
def_exit_tx 383
```

Joe the Mortal will have an RxLevel: Real and a TxLevel: Real Jack the Malk, who likes to walk around Obfuscated and has Obfuscate 2 will have an RxLevel: Real (he sees what the mortals see) but a TxLevel: Obf2 Jimmy the Nossie, who is using the Mask and has Obfuscate 4, but doesn't try to make himself invisible will have an RxLevel: Real (as Jack) and a TxLevel: Real Obf4. He will also set his @desc to what the mortals see and &OBFDESC to his real slimy desc. Simply put, he will be visible to mortals, but not with his real desc.

Aldrin the Gangrel, has Auspex 4 and activates it. Therefore, his TxLevel will still be Real, but RxLevel: Real Obf1 Obf2 Obf3 Obf4 (all of them). So he can see Joe, Jack and Jimmy's both descs. Joe, on the other hand, won't see Jack at all. He will still see Jimmy, but only Jimmy's @desc, not the OBFDESC

Frida the Fae... will have RxLevel: Real Fae and TxLevel: Real Fae. @desc set to the mundane desc, &FAEDESC set to the Chimerical desc. Emily the Enchanted will have an RxLevel: Real Fae, but the TxLevel: Real. No &FAEDESC on her, although she'll be able to see it the one on Frida. Gil the Garou, while travelling through the Umbra, will have RxLevel: Umbra and TxLevel: Umbra. &UMBRADESC is his friend. He won't see mortals or other characters who are not in the Umbra.

Barbie the Bastet, who's only peeking in the Umbra, without going there, will have RxLevel: Umbra, TxLevel: Real. Dangerous position since she can't see the things that see her.

Deanna the Drake, who activates her spirit vision, will have RxLevel: Real Umbra and TxLevel: Real. She will see characters in Umbra and real world at the same time and perceive the desc appropiate to the realm the ohter character is in.

Wanda the Wraith: RxLevel: Real Shadow, TxLevel: Shadow. Her @desc would be empty, but the &SHADOWDESC should be set. Marie the Mortal+ Medium: RxLevel: Real Shadow, TxLevel: Real

Global code objects that all characters should be able to use:

```
RxLevel: All, TxLevel: All
```

10.3.9 Example 3: Softcode

Considering the config directives from example 2 and assuming a function getstat(<dbref>,<stat>) that will return the value of a player's stat from the sheet here are softcode examples that implement some of the WoD powers. In a real game you would have to use some more checks, of course.

```
@create Reality Levels Commands (RLS)
&CMD_OBFON rls=$+obf/on:@switch [setr(0, getstat(%#,Obfuscate))]=0, @pemit %#=You don
↪'t have Obfuscate!, {@txlevel %#=!All Obf%q0; @pemit %#=You are now invisible.}
&CMD_OBFOFF rls=$+obf/off:@txlevel %#=Real; @pemit %#=You are now visible.
@@ Note: +obf/on clears all TxLevels before setting the appropiate Obf
```

(continues on next page)

(continued from previous page)

```
@@ This is necesary, because a character might advance from Obf2 to
@@ Obf3 and he should be visible /only/ on the Obf3 level.
@@ +obf/off simply sets the Real Tx level, without clearing the Obf. The
@@ reason is the Mask. Players with Obf3 or higher who use the Mask should
@@ +obf/on, then +obf/off after approval and everything is set.
&CMD_AUSPEXON rls=$+auspex/on:@switch [setr(0, getstat(%#, Auspex))]=0, @pemit %#=You␣
↪don't have Auspex!, {@rxlevel %#=[iter(lnum(1, %q0), Obf##)]; @pemit %#=Auspex␣
↪enabled.}
&CMD_AUSPEXOFF rls=$+auspex/off:@switch [hasrxlevel(%#, Obf1)]=0, @pemit %#= You don
↪'t have Auspex enabled!, {@rxlevel %#=[iter(lnum(1, 5), !Obf##)]; @pemit %#=Auspex␣
↪disabled.}
&CMD_UMBRAENTER rls=$+umbra/enter:@rxlevel %#=!Real Umbra; @txlevel %#= !Real Umbra;␣
↪@pemit %#=You are now in the Umbra.
&CMD_UMBRALEAVE rls=$+umbra/leave:@rxlevel %#=Real !Umbra; @txlevel %#= Real !Umbra;␣
↪@pemit %#=You left the Umbra.
&CMD_PEEKON rls=$+peek/on:@switch hastxlevel(%#,Umbra)=1, {@rxlevel %#=Real !Umbra;␣
↪@pemit %#=You are now peeking in the real world}, {@rxlevel %#=!Real Umbra; @pemit %
↪#=You are now peeking into the Umbra}
&CMD_PEEKOFF rls=$+peek/off:@rxlevel %#=!Real !Umbra [setinter(Real Umbra, txlevel(%
↪#))]; @pemit %#=You are no longer peeking.
```

10.4 Execscript and external programs and scripts

10.4.1 Execscript variables

Execscript Built in variables

Variable	Description
MUSH_PLAYER	player dbref#
MUSH_CAUSE	cause dbref#
MUSH_CALLER	caller dbref#
MUSH_OWNER	owner of player dbref#
MUSH_FLAGS	space delimited list of flags on player
MUSH_TOGGLES	space delimited list of toggles on player
MUSH_OFLAGS	space delimited list of flags of player owner
MUSH_OTOGGLES	space delimited list of toggles of player owner
MUSHL_VARS	space delimited list of MUSH attributes from player This is passed from the mush's EXECSCRIPT_VARS attr

Execscript Dynamic variables

Variable	Description
MUSHV_<arg>	<arg> variable passed from MUSHL_VARS These are the attributes from EXECSCRIPT_VARS

Execscript Register variables

Variable	Description
MUSHQ_<arg>	setq registers 0-9 and a-z
MUSHQN_<arg>	labels that are assigned the setq vars
MUSHN_<arg>	The labels that were defined by any register Note: they must be ASCII-7 clean and contain no white spaces

10.4.2 Execsript set object

The script executed with execscript() will read in a file with the same name as the script ending in '.set'. This is a loader object that will set attributes and registers back into the mush that you wish to pass from the script. The fields are SPACE SEPARATED. The values are NOT evaluated.

Execscript set object field format

Execscript set object Dynamic variables

```
VARNAME         OWNER         CONTENTS (or leave null to clear)
```

Execscript set object Dynamic variables Examples

```
SEX #123 Male
DESC #123 %r%tThis is a willow tree of unique description%r%rIt sways in the wind.
RED #123 This is the color %ch%crred%cn.
WIPETHISATTR #123
MULTILINE #123 This is a line
that continues on
because of the line feed (a control-M) on each line
on the lines above
```

Execscript set object Register variables

```
REGISTER        Q             CONTENTS (or leave null to clear)
```

Execscript set object Register variables Examples

Note: The last exammple clears register 0

```
W Q This is stored in register W
1 Q This is stored in register 1
0 Q
foo QN this sets register with label 'foo'
```

10.4.3 Execscript Example bash script

```
#!/bin/bash
echo "This was called by player ${MUSH_PLAYER} that is owned by ${MUSH_OWNER}"
echo "Displaying Registers:"
regs="0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z"
for list in ${regs}
do
    eval echo "Register ${list}: \${MUSHQ_${list}}"
done
echo "Displaying variables:"
for vars in ${MUSHL_VARS}
do
    eval echo "Variable ${vars}: \${MUSHV_${vars}}"
done
```

10.4.4 Exescript Notes and warnings

While MUSHL_VARS are sanitized on what is allowable as a mush variable, this is not necessarilly sanitized on what the calling script can fetch as a valid variable. Of note, you can not set environment variables that contain an equals sign. Be aware of this limitation.

Remember, MUSHL_VARS is the environment variable seen by the script. This is EXECSCRIPT_VARS on the mush itself, that is the attribute set on the target that contains the execscript() that is being executed.

10.4.5 Scripts to be used with execscript

```
account/                        -- Directory for execscripts relating to account␣
↪creation
compile39.sh                    -- Script for patching and compiling RhostMUSH 3.9
compile.sh                      -- Script for patching and compiling RhostMUSH
config.sh                       -- Script for setting compile time options for␣
↪RhostMUSH
debug.sh                        -- Script for debugging RhostMUSH
dict.sh                         -- Script for querying a dictionary
diff.sh                         -- Script for querying differences between two␣
↪arguments
fortune.sh                      -- Script for querying fortune program
fullweather.sh                  -- Script for querying a graphical weather forecast␣
↪(alternative)
git.sh                          -- Script for querying git version of RhostMUSH
hello.sh                        -- Script for teaching execscript for 'Hello World'
iostat.sh                       -- Script for querying server stats of RhostMUSH
jsonvalidate.sh                 -- Python Script for validating JSON
logsearch.sh                    -- Script for searching throgh logfiles for RhostMUSH
math_example.sh                 -- Examples of math operations to be used with math.sh
math.sh                         -- Script for mathematical operations
memory.sh                       -- Script for querying memory usage of RhostMUSH
mkindx.sh                       -- Script for indexing RhostMSH helpfiles
pastebinread.sh                 -- Script for reading data from a pastebin URL
pastebinwrite.sh                -- Script for writing data to a pastebin
qspell.sh                       -- Script for checking spelling (alternative)
quota.sh                        -- Script for checking disk quote and usage
random.sh                       -- Script for getting a random number
```

(continues on next page)

(continued from previous page)

```
roomlog.sh                      -- Script for viewing logs in roomlog directory
spell.sh                        -- Script for checking spelling
stats.sh                        -- Script for querying server and process stats for␣
↪RhostMUSH
thes.sh                         -- Script for adding a word to the dictionary for␣
↪spell scripts
tinyurl.sh                      -- Script for shortening a URL
weather.sh                      -- Script for querying a graphical weather forecast
web.sh                          -- Script for querying an arbitary website
```

10.5 Using printf() for advanced text output

The function printf() in Rhost can be used to greatly reduce coding in efforts for outputs, screens and data display. It can automatically center, justify and wrap the text parameters given to it.

10.5.1 Printf Example one

```
@emit printf(|$-12s|$12s|$^12s$&14s$_12s|,a b c, d e f, g h i, wrap(lnum(20),12, l, |,
↪ |), j k l)

|a b c       |       d e f|   g h i    |0 1 2 3 4 5 |j     k    l|
                                        |6 7 8 9 10  |
                                        |11 12 13 14 |
                                        |15 16 17 18 |
                                        |19          |
```

10.5.2 Printf Example two

```
@emit printf($14&s $^4&s $-3&s $15&s,
iter(Bruised|Hurt|Injured|Wounded|Mauled|Crippled|Incapacitated,##,|,%R),
iter(|-1|-1|-2|-2|-5|,##,|,%r),iter(lnum(1,7),%[[if(gte(get(%#/damage),##),X,%b)]%],,
↪%r),
* Aggravated%RX Lethal%R/ Bashing)

         Bruised      [ ]    * Aggravated
            Hurt  -1  [ ]        X Lethal
         Injured  -1  [ ]       / Bashing
         Wounded  -2  [ ]
          Mauled  -2  [ ]
        Crippled  -5  [ ]
   Incapacitated      [ ]
```

10.5.3 Printf Example three

```
@emit [printf($-10|"'s$-60|"s,a b c d e f g h i j k l m n o p q r s t u v w x y z,
this is a test a groovy test blah blah blah [repeat(blah%b,100)])]END!

a b c d e this is a test a groovy test blah blah blah blah blah blah
f g h i j blah blah blah blah blah blah blah blah blah blah blah blah
k l m n o blah blah blah blah blah blah blah blah blah blah blah blah
p q r s t blah blah blah blah blah blah blah blah blah blah blah blah
u v w x y blah blah blah blah blah blah blah blah blah blah blah blah
z         blah blah blah blah blah blah blah blah blah blah blah blah
blah blah blah blah blah blah blah blah blah blah blah blah blah blah
blah blah blah blah blah blah blah blah blah blah blah blah blah blah
blah blah blah blah blah blah blah                                  END!
```

CHAPTER

ELEVEN

FORMAT FOR IMAGE FILES

The image format goes like this:

Data Type	Example	Description
INT	3	TYPE: room 0, thing 1, exit 2, player 3, zone 4, garbage 5
STRING	Wizard	NAME: of the target. Verbatum, no quotes surround it
*INT	123	LOCATION: dbref# of the target. No prepending '#' used.
*INT	234	CONTENTS: The first content in a linked list content table (-1 if none)
*INT	345	EXITS: The first exit in a linked list exit table (-1 if none)
*INT	0	LINK: This is the 'home' of the object or what it's linked to (-1 for none)
*INT	123	NEXT: The next thing after this item for a content holder
STRING	#123	LOCK: The boolean string lock if it exists. (empty if no lock)
*INT	1	OWNER: The dbref# owner of the target. For players same dbref as player.
INT	789	PARENT: The parent of the target. (-1 if none)
*INT	99999	MONEY: The int value of the money the players has.
INT	194592	FLAGS1: The first word of flags (@set flags) on a player (see FLAGS)
INT	194222	FLAGS2: The second word of flags (@set flags) on a player (see FLAGS)
INT	199999	FLAGS3: The third word of flags (@set flags []) on a player (see FLAGS)
INT	1582958	FLAGS4: The forth word of flags (@set flags []) on a player (see FLAGS)
INT	159955	TOGGLES1: The first word of toggles (@toggle) on a player (see TOGGLES)
INT	159958	TOGGLES2: The second word of toggles (@toggle) on a player (see TOGGLES)
INT	159958	POWER1: The first word of powers (@power) on a player (see POWERS)
INT	159958	POWER2: The second word of powers (@power) on a player (see POWERS)
INT	159958	POWER3: The third word of powers (@power) on a player (see POWERS)
INT	159958	DEPOWER1: The first word of depowers (@depower) on a player (see DEPOWERS)
INT	159958	DEPOWER2: The second word of depowers (@depower) on a player (see DEPOWERS)
INT	159958	DEPOWER3: The third word of depowers (@depower) on a player (see DEPOWERS)
INT	-1	ZONE(S): The list of zones starting here and ending with '-1'. (see ZONES)
>STRING	>VA	ATTRIBUTENAME: Attribute name to store, starts with > identifier
STRING	Wheee	ATTRIBUTECONTENTS: Contents of attribute. Multi-lines seperate with ^M (control-M)
>STRING	>Desc	ATTRIBUTENAME: Another attribute to chain in
STRING	Ugly	ATTRIBUTECONTENTS: Contents of the next attribute
>STRING	*Password	PASSWORDATTRIB: Special password attribute. Attribute name is '*Password'
STRING	6xy$xy	PASSWORDCONTENTS: The SHA512 password (if glibc 2.7+ supported on system) (see PASS)
<	<	This is the marker to specify the end of the attribute contents. This is always the last line

Note: Any Data type starting with '*' is ignored when @snapshot/loading.

The structure above with the examples would look like this in the file:

```
3
Wizard
123
234
345
0
123
#123
1
789
99999
194592
194222
199999
1582958
159955
159958
159958
159958
159958
159958
159958
159958
-1
>VA
Wheee
>Desc
Ugly
>*Password
$6$xy$xy
<
```

11.1 HELP key indexes for the values

FLAGS: The following flags are to be used. They are BITWISE masks that you need to add together for the values that you apply:

```
/* First word of flags */
#define SEETHRU         0x00000008      /* Can see through to the other side */
#define WIZARD          0x00000010      /* gets automatic control */
#define LINK_OK         0x00000020      /* anybody can link to this room */
#define DARK            0x00000040      /* Don't show contents or presence */
#define JUMP_OK         0x00000080      /* Others may @tel here */
#define STICKY          0x00000100      /* Object goes home when dropped */
#define DESTROY_OK      0x00000200      /* Others may @destroy */
#define HAVEN           0x00000400      /* No killing here, or no pages */
#define QUIET           0x00000800      /* Prevent 'feelgood' messages */
#define HALT            0x00001000      /* object cannot perform actions */
#define TRACE           0x00002000      /* Generate evaluation trace output */
#define GOING           0x00004000      /* object is available for recycling */
#define MONITOR         0x00008000      /* Process ^x:action listens on obj? */
#define MYOPIC          0x00010000      /* See things as nonowner/nonwizard */
#define PUPPET          0x00020000      /* Relays ALL messages to owner */
#define CHOWN_OK        0x00040000      /* Object may be @chowned freely */
#define ENTER_OK        0x00080000      /* Object may be ENTERed */
```

(continues on next page)

(continued from previous page)

```
#define VISUAL          0x00100000      /* Everyone can see properties */
#define IMMORTAL        0x00200000      /* Object can't be killed */
#define HAS_STARTUP     0x00400000      /* Load some attrs at startup */
#define OPAQUE          0x00800000      /* Can't see inside */
#define VERBOSE         0x01000000      /* Tells owner everything it does. */
#define INHERIT         0x02000000      /* Gets owner's privs. (i.e. Wiz) */
#define NOSPOOF         0x04000000      /* Report originator of all actions. */
#define ROBOT           0x08000000      /* Player is a ROBOT */
#define SAFE            0x10000000      /* Need /override to @destroy */
#define CONTROL_OK      0x20000000      /* ControlLk specifies who ctrls me */
#define HEARTHRU        0x40000000      /* Can hear out of this obj or exit */
#define TERSE           0x80000000      /* Only show room name on look */

/* Second word of flags */
#define KEY             0x00000001      /* No puppets */
#define ABODE           0x00000002      /* May @set home here */
#define FLOATING        0x00000004      /* Inhibit Floating room.. msgs */
#define UNFINDABLE      0x00000008      /* Cant loc() from afar */
#define PARENT_OK       0x00000010      /* Others may @parent to me */
#define LIGHT           0x00000020      /* Visible in dark places */
#define HAS_LISTEN      0x00000040      /* Internal: LISTEN attr set */
#define HAS_FWDLIST     0x00000080      /* Internal: FORWARDLIST attr set */
#define ADMIN           0x00000100      /* Player has admin privs */
#define GUILDOBJ        0x00000200
#define GUILDMASTER     0x00000400      /* Player has gm privs */
#define NO_WALLS        0x00000800      /* So to stop normal walls */
#define REQUIRE_TREES   0x00001000      /* Trees are required on this target for␣
↪attrib sets */
/* ----FREE----         0x00002000 */   /* #define OLD_NOROBOT  0x00002000 */
#define SCLOAK          0x00004000
#define CLOAK           0x00008000
#define FUBAR           0x00010000
#define INDESTRUCTABLE  0x00020000      /* object can't be nuked */
#define NO_YELL         0x00040000      /* player can't @wall */
#define NO_TEL          0x00080000      /* player can't @tel or be @tel'd */
#define FREE            0x00100000      /* object/player has unlim money */
#define GUEST_FLAG      0x00200000
#define RECOVER         0x00400000
#define BYEROOM         0x00800000
#define WANDERER        0x01000000
#define ANSI            0x02000000
#define ANSICOLOR       0x04000000
#define NOFLASH         0x08000000
#define SUSPECT         0x10000000      /* Report some activities to wizards */
#define BUILDER         0x20000000      /* Player has architect privs */
#define CONNECTED       0x40000000      /* Player is connected */
#define SLAVE           0x80000000      /* Disallow most commands */

/* Third word of flags - Thorin 3/95 */
#define NOCONNECT       0x00000001
#define DPSHIFT         0x00000002
#define NOPOSSESS       0x00000004
#define COMBAT          0x00000008
#define IC              0x00000010
#define ZONEMASTER      0x00000020
#define ALTQUOTA        0x00000040
#define NOEXAMINE       0x00000080
```

(continues on next page)

(continued from previous page)

```
#define NOMODIFY        0x00000100
#define CMDCHECK        0x00000200
#define DOORRED         0x00000400
#define PRIVATE         0x00000800      /* For exits only */
#define NOMOVE          0x00001000
#define STOP            0x00002000
#define NOSTOP          0x00004000
#define NOCOMMAND       0x00008000
#define AUDIT           0x00010000
#define SEE_OEMIT       0x00020000
#define NO_GOBJ         0x00040000
#define NO_PESTER       0x00080000
#define LRFLAG          0x00100000
#define TELOK           0x00200000
#define NO_OVERRIDE     0x00400000
#define NO_USELOCK      0x00800000
#define DR_PURGE        0x01000000      /* For rooms only...internal */
#define NO_ANSINAME     0x02000000      /* Remove the ability to set @ansiname */
#define SPOOF           0x04000000
#define SIDEFX          0x08000000      /* Allow enactor to use side-effects */
#define ZONECONTENTS    0x10000000      /* Search contents of zonemaster for
↪$commands */
#define NOWHO           0x20000000      /* Player in WHO doesn't show up - use with␣
↪@hide */
#define ANONYMOUS       0x40000000      /* Player set shows up as 'Someone' when␣
↪talking */
#define BACKSTAGE       0x80000000      /* Immortal toggle for items on control */

/* Forth word of flags - Thorin 3/95 */
#define NOBACKSTAGE     0x00000001      /* Immortal toggle to control no-backstage */
#define LOGIN           0x00000002      /* Enable player to login past @disable␣
↪logins */
#define INPROGRAM       0x00000004      /* Player is inside a program */
#define COMMANDS        0x00000008      /* Optional define for $commands */
#define MARKER0         0x00000010      /* TM 3.0 marker flags */
#define MARKER1         0x00000020
#define MARKER2         0x00000040
#define MARKER3         0x00000080
#define MARKER4         0x00000100
#define MARKER5         0x00000200
#define MARKER6         0x00000400
#define MARKER7         0x00000800
#define MARKER8         0x00001000
#define MARKER9         0x00002000
#define BOUNCE          0x00004000      /* That lovly TM 3.0 Bouncey thingy */
#define SHOWFAILCMD     0x00008000      /* Show failed $commands defauilt error */
#define NOUNDERLINE     0x00010000      /* Strip UNDERLINE character from ANSI */
#define NONAME          0x00020000      /* Target does not display name with look */
#define ZONEPARENT      0x00040000      /* Target zone allows inheritance of attribs␣
↪*/
#define SPAMMONITOR     0x00080000      /* Monitor the target for spam */
#define BLIND           0x00100000      /* Exits and locations snuff arrived/left */
#define NOCODE          0x00200000      /* Players may not code */
#define HAS_PROTECT     0x00400000      /* Player target has protect name data */
#define XTERMCOLOR      0x00800000      /* Extended AnSI Xterm colors */
#define HAS_ATTRPIPE    0x01000000      /* Attribute piping via @pipe */
/* 0x02000000 free */
```

(continues on next page)

(continued from previous page)

```
/* 0x04000000 free */
/* 0x08000000 free */
/* 0x10000000 free */
/* 0x20000000 free */
/* 0x40000000 free */
/* 0x80000000 free */
```

TOGGLES: Toggles are BITWISE masks taht need to be applied for each word like the flags above. They are added together for each word type:

```
/* First word of toggles - Thorin 3/95 */
#define TOG_MONITOR             0x00000001      /* Active monitor on player */
#define TOG_MONITOR_USERID      0x00000002      /* show userid */
#define TOG_MONITOR_SITE        0x00000004      /* show site */
#define TOG_MONITOR_STATS       0x00000008      /* show stats */
#define TOG_MONITOR_FAIL        0x00000010      /* show fails */
#define TOG_MONITOR_CONN        0x00000020
#define TOG_VANILLA_ERRORS      0x00000040      /* show normal error msg */
#define TOG_NO_ANSI_EX          0x00000080      /* supress ansi stuff in ex */
#define TOG_CPUTIME             0x00000100      /* show cpu time for cmds */
#define TOG_MONITOR_DISREASON   0x00000200
#define TOG_MONITOR_VLIMIT      0x00000400
#define TOG_NOTIFY_LINK         0x00000800
#define TOG_MONITOR_AREG        0x00001000
#define TOG_MONITOR_TIME        0x00002000
#define TOG_CLUSTER             0x00004000      /* Object is part of a cluster */
#define TOG_SNUFFDARK           0x00008000      /* Snuff Dark Exit Viewing */
#define TOG_NOANSI_PLAYER       0x00010000      /* Do not show ansi player names */
#define TOG_NOANSI_THING        0x00020000      /* ... things */
#define TOG_NOANSI_ROOM         0x00040000      /* ... rooms */
#define TOG_NOANSI_EXIT         0x00080000      /* ... exits */
#define TOG_NO_TIMESTAMP        0x00100000      /* Do not modify timestamps on target␣
↪*/
#define TOG_NO_FORMAT           0x00200000      /* Override @conformat/@exitformat */
#define TOG_ZONE_AUTOADD        0x00400000      /* Automatically add FIRST zone in␣
↪list */
#define TOG_ZONE_AUTOADDALL     0x00800000      /* Automatically add ALL zones in␣
↪list */
#define TOG_WIELDABLE           0x01000000      /* Marker to specify if object is␣
↪wieldable */
#define TOG_WEARABLE            0x02000000      /* Marker to specify if object is␣
↪wearable */
#define TOG_SEE_SUSPECT         0x04000000      /* Specify who sees suspect in WHO/
↪MONITOR */
#define TOG_MONITOR_CPU         0x08000000      /* Specify who sees CPU overflow␣
↪allerts */
#define TOG_BRANDY_MAIL         0x10000000      /* Define brandy like mail interface␣
↪*/
#define TOG_FORCEHALTED         0x20000000      /* The item toggled can @force halted␣
↪things */
#define TOG_PROG                0x40000000      /* Can use @program on other people/
↪things */
#define TOG_NOSHELLPROG         0x80000000      /* Target can not issue commands␣
↪inside a prog */

/* Second word of toggles -- Ash */
```

(continues on next page)

(continued from previous page)

```
#define TOG_EXTANSI             0x00000001      /* Specify if target can used␣
↪extended ansi naming */
#define TOG_IMMPROG             0x00000002      /* Only an immortal can @quitprogram␣
↪them */
#define TOG_MONITOR_BFAIL       0x00000004      /* Monitor if a failed connect on bad␣
↪character */
#define TOG_PROG_ON_CONNECT     0x00000008      /* Reverse logic of program on␣
↪connect */
#define TOG_MAIL_STRIPRETURN    0x00000010      /* Strip carrage return in mail␣
↪combining */
#define TOG_PENN_MAIL           0x00000020      /* Use PENN style syntax */
#define TOG_SILENTEFFECTS       0x00000040      /* Silents did_it() functionality on␣
↪target */
#define TOG_IGNOREZONE          0x00000080      /* Target is set to @icmd zones */
#define TOG_VPAGE               0x00000100      /* Target sees alias in pages */
#define TOG_PAGELOCK            0x00000200      /* Target issues pagelocks as normal␣
↪*/
#define TOG_MAIL_NOPARSE        0x00000400      /* Don't parse %t/%b/%r in mail */
#define TOG_MAIL_LOCKDOWN       0x00000800      /* Mortal-accessed mail/number and␣
↪mail/check */
#define TOG_MUXPAGE             0x00001000      /* Have 'page' work like MUX */
#define TOG_NOZONEPARENT        0x00002000      /* Zone Child does NOT inherit parent␣
↪attribs */
#define TOG_ATRUSE              0x00004000      /* Enactor can use Attribute based␣
↪USELOCKS */
#define TOG_VARIABLE            0x00008000      /* Set exit to be variable */
#define TOG_KEEPALIVE           0x00010000      /* Send 'keepalives' to the target␣
↪player */
#define TOG_CHKREALITY          0x00020000      /* Target checks @lock/user for␣
↪Reality passes */
#define TOG_NOISY               0x00040000      /* Always do noisy sets */
#define TOG_ZONECMDCHK          0x00080000      /* Zone commands checked on target␣
↪like @parent */
#define TOG_HIDEIDLE            0x00100000      /* Allow wizards/immortals to hide␣
↪their idle time */
#define TOG_MORTALREALITY       0x00200000      /* Override the wiz_always_real␣
↪setting */
#define TOG_ACCENTS             0x00400000      /* Accents being displayed */
#define TOG_PREMAILVALIDATE     0x00800000      /* Pre-Validate the mail send list␣
↪before sending mail */
#define TOG_SAFELOG             0x01000000      /* Allow 'clean logging' by the␣
↪player */
#define TOG_UTF8                0x02000000      /* UTF8 being displayed */
/* 0x04000000 free */
#define TOG_NODEFAULT           0x08000000      /* Allow target to inherit default␣
↪attribs */
#define TOG_EXFULLWIZATTR       0x10000000      /* Examine Wiz attribs */
#ifdef ENH_LOGROOM
#define TOG_LOGROOMENH          0x20000000      /* Enhanced Room Logging */
#endif
#define TOG_LOGROOM             0x40000000      /* Log Room's location/contents */
#define TOG_NOGLOBPARENT        0x80000000      /* Target does not inherit global␣
↪attributes */
```

POWERS: Powers are handled a bit differently. They're used as BITWISE shift markers that you would have to compute the shift then add it after the fact.:

```
/* First word of power positions.  Each position is 2 bits so the
   number here is how far over to shift the 2 bit pattern         */
#define POWER_CHANGE_QUOTAS     0
#define POWER_CHOWN_ME          2
#define POWER_CHOWN_ANYWHERE    4
#define POWER_CHOWN_OTHER       6
#define POWER_WIZ_WHO           8
#define POWER_EX_ALL            10
#define POWER_NOFORCE           12
#define POWER_SEE_QUEUE_ALL     14
#define POWER_FREE_QUOTA        16
#define POWER_GRAB_PLAYER       18
#define POWER_JOIN_PLAYER       20
#define POWER_LONG_FINGERS      22
#define POWER_NO_BOOT           24
#define POWER_BOOT              26
#define POWER_STEAL             28
#define POWER_SEE_QUEUE         30

/* Second word of power positions. */
#define POWER_SHUTDOWN          0
#define POWER_TEL_ANYWHERE      2
#define POWER_TEL_ANYTHING      4
#define POWER_PCREATE           6
#define POWER_STAT_ANY          8
#define POWER_FREE_WALL         10
#define POWER_EXECSCRIPT        12
#define POWER_FREE_PAGE         14
#define POWER_HALT_QUEUE        16
#define POWER_HALT_QUEUE_ALL    18
#define POWER_FORMATTING        20
#define POWER_NOKILL            22
#define POWER_SEARCH_ANY        24
#define POWER_SECURITY          26
#define POWER_WHO_UNFIND        28

/* Third word of power positions. */
#define POWER_OPURGE            0
#define POWER_HIDEBIT           2
#define POWER_NOWHO             4
#define POWER_FULLTEL_ANYWHERE  6
#define POWER_EX_FULL           8
#define POWER_API               10
#define POWER_MONITORAPI        12
#define POWER_WIZ_IDLE          14
#define POWER_WIZ_SPOOF         16
/* 18 free */
/* 20 free */
/* 22 free */
/* 24 free */
/* 26 free */
/* 28 free */
/* 30 free */
```

DEPOWERS: like @powers they are handled with a BITWISE offshift that you will have to calculate then add:

```
/* First word */
```

(continues on next page)

(continued from previous page)

```
#define DP_WALL                 0
#define DP_LONG_FINGERS         2
#define DP_STEAL                4
#define DP_CREATE               6
#define DP_WIZ_WHO              8
#define DP_CLOAK                10
#define DP_BOOT                 12
#define DP_PAGE                 14
#define DP_FORCE                16
#define DP_LOCKS                18
#define DP_COM                  20
#define DP_COMMAND              22
#define DP_MASTER               24
#define DP_EXAMINE              26
#define DP_NUKE                 28
#define DP_FREE                 30

/* Second word */
#define DP_OVERRIDE             0
#define DP_TEL_ANYWHERE         2
#define DP_TEL_ANYTHING         4
#define DP_PCREATE              6
#define DP_POWER                8
#define DP_QUOTA                10
#define DP_MODIFY               12
#define DP_CHOWN_ME             14
#define DP_CHOWN_OTHER          16
#define DP_ABUSE                18
#define DP_UNL_QUOTA            20
#define DP_SEARCH_ANY           22
#define DP_GIVE                 24
#define DP_RECEIVE              26
#define DP_NOGOLD               28
#define DP_NOSTEAL              30
/* Third word...and there was much rejoicing */
#define DP_PASSWORD             0
#define DP_MORTAL_EXAMINE       2
#define DP_PERSONAL_COMMANDS    4
/* 6  free */
#define DP_DARK                 8
/* 10 free */
/* 12 free */
/* 14 free */
/* 16 free */
/* 18 free */
/* 20 free */
/* 22 free */
/* 24 free */
/* 26 free */
/* 28 free */
/* 30 free */
```

ZONES: Zones are special. If there are no zones, the value will be '-1'.

So entering zones if there are no zones:

```
-1
```

Entering zones if it has three zones (#123, #456, and #789):

```
123
456
789
-1
```

As you see, the last value of the zone *MUST* be -1. This tells it that there are no more zones to add.

CHAPTER

TWELVE

REQUIREMENTS IF USING WINDOWS

- (BETA ONLY) cygwin under Windows. It requires the entire base development set and Requirements below.

12.1 Installing on Windows 10 with BASH

Rhost can be compiled and run under the new Bash on Ubuntu on Windows. This has been tested with the Preview build 14342.

1. After installing Bash you will need to install the following packages:
    - gcc
    - git
    - make
    - libpcre3 (optional)
    - libpcre3-dev (optional)
    - openssl (optional)

2. When configuring rhost (using confsource) select the Disable Debugmon option.

12.2 Installing on Windows with Cygwin

Rhost does work under windows using the cygwin package.

1. When you do install cygwin, the following packages must be added:
    - bash
    - crypt
    - gcc
    - gdbm
    - git
    - make
    - openssl (optional)
2. The src/Makefile has to manually have the CYGWIN line uncommented.

12.3 Startig RhostMUSH on Windows

When you issue Startmush, you must pass it the -cyg option.

CHAPTER

THIRTEEN

UPGRADING A LEGACY RHOSTMUSH INSTALLATION

13.1 Converting database betwen GDBM and QDBM

Ok, if you plan to recompile your game that is using GDBM to QDBM, or visa versa some bad news.

The databases are NOT COMPATIBLE to each other, at least in the binary form.

13.1.1 Downgrading QDBM to GDBM

> **Warning:** I would NEVER change from QDBM back to GDBM, but if you're set on it these steps:
>
> You would use the same steps if you plan to move QDBM to GDBM. I however would not do this. Moving from QDBM to GDBM is a huge step backwards. Seriously, don't do it unless you have absolutely no other recourse.
>
> IF you plan (for whatever reason) to move from QDBM to GDBM, you should verify the following

1. You have on a 64 bit system, no object that has more than 400 attributes on it.
2. You have on a 32 bit system, no object that has more than 750 attributes on it.
3. Any CONTENT of any attribute must be below 4000 characters in length.
4. Once you have that done, you may follow the procedures below on converting (upgrade) from GDBM to QDBM. This works the same as converting (downgrading) QDBM back down to GDBM

13.1.2 Upgradging GDBM to QDBM

Now, if you've kept reading and plan to convert your GDBM database to QDBM great news! It's more stable, it's faster, and lets you have far more flexibility.

So, BEFORE YOU RECOMPILE YOUR CODE. This is what you have to do.

While logged in to your mush, issue the following commands

A. @dump/flat – This will make a flatfile dump of your MUSH database

B. wmail/unload – This will make a flatfile dump of your MAIL database

C. @areg/unload – If you use the AutoRegistration engine, this dumps it

D. newsdb/unload – If you use the hardcoded news/bbs engine. This dumps it

Verify the files exist

A. Server/game/data/netrhost.db.flat

B. Server/game/data/RhostMUSH.dump.folder

C. Server/game/data/RhostMUSH.dump.mail

D. (Optional) Server/game/data/RhostMUSH.areg.dump

E. (Optional) Server/game/data/RhostMUSH.news.flat

Shutdown the MUSH

@shutdown your mush

From the Server directory

A. make clean

B. make confsource 1. Select QDBM and if you wish at this time increase your LBUF size 2. Select any other options you may want

C. (r)un and let it compile.

D. Main DB: Delete (rm) the following files (from Rhost/Server/game/data):

```
netrhost.gdbm*
netrhost.db
netrhost.db.new
netrhost.db.new.prev
```

E. Mail DB: Delete (rm) the following files (from Rhost/Server/game/data):

```
RhostMUSH.folder.dir
RhostMUSH.folder.pag
RhostMUSH.mail.dir
RhostMUSH.mail.pag
```

F. (Optional) AutoReg DB: Delete (rm) the following files (from Rhost/Server/game/data):

```
RhostMUSH.areg.dir
RhostMUSH.areg.pag
```

G. (Optional) News/BBS DB: Delete (rm) the following files (from Rhost/Server/game/data):

```
RhostMUSH.news.dir
RhostMUSH.news.pag
```

From the Server/game directory

A. Load the database:

```
./db_load data/netrhost.gdbm data/netrhost.db.flat data/netrhost.db.new
```

B. Start the MUSH back:

```
./Startmush
```

While logged into the mush issue the following commands

A. Load in the mail database:

```
wmail/load
```

B. (optional) Load in the autoreg database:

```
@areg/load
```

C. (optional) Load in the news/bbs database:

```
newsdb/load
```

Verify that you have QDBM running and your valid values

A. @list options system

B. @list options (spammy)

13.2 Updating RhostMUSH prior to 3.9.5p2

Ok.

So you're running an old RhostMUSH.

One prior to 3.9.5p2 and want to take advantage of the new format of the Makefile and the automated mysql stuff and all the other goodies that isn't really (easilly) done with just patch.sh.

Well, you're in luck. It is actually fairly easy to do.

This is what you have to do.

First thing's first.

1. Log into your existing mush. Let's make current backups of all your flatfiles. Issue:

```
@dump/flat
wmail/unload
@areg/unload
newsdb/unload
```

2. Shutdown your game:

```
@shutdown
```

3. Make an image of all your current backed up files. From The Server/game directory you would type:

```
./backup_flat.sh -s
```

Note: Please remember the '-s' to the ./backup_flat.sh.

4. Make note of the most recently created file in the directory Server/game/oldflat. It's usually named something like:

```
RhostMUSH.dbflat1.tar.gz
```

Note: You will need this file later.

5. Rename your 'Rhost' directory to something else. This is the directory that you have containing the 'Server' directory. Name it anything you want other than 'Rhost'. For those not used to unix you would type:

```
mv Rhost Rhost_old
```

6. Pull in the latest Rhost. You would type:

```
git clone https://github.com/RhostMUSH/trunk Rhost
```

Note: You would type this in the same directory you have renamed your old 'Rhost' directory

7. go into the Rhost/Server directory. Type:

```
make confsource
```

   Select what options you want (including the mysql and other goodies) then compile it (also within the menu, it's the 'r' option).

8. Once your game is compiled and ready to go you need to copy in the data from your old game. Copy the RhostMUSH.dbflat1.tar.gz we mentioned in step #4 to the Server/game directory of your NEW GAME DIRECTORY. From within the 'game' directory of your current game you should be able to issue (if you named the old one Rhost_old). Again this needs to be done FROM YOUR Server/game directory!!!

   A. cp netrhost.conf netrhost.conf.orig

   B. cp ../../Rhost_old/Server/game/RhostMUSH.dbflat1.tar.gz .

   C. tar -zxvf RhostMUSH.dbflat1.tar.gz

   D. Compare your current netrhost.conf to the default one that came with the source (that you renamed to netrhost.conf.orig). Likely the only sections you have to add to your current netrhost.conf (that came with your RhostMUSH.dbflat1.tar.gz archive), will be toward the end, with the include rhost_ingame.conf and rhost_mysql.conf. Depending on how old your game is coming from you may need to add more options. Any config option that is the same between the netrhost.conf files do not have to be copied over, and you want to keep your custom settings (like don't port or other stuff you have already customized).

   E. Load in your flatfile information:

```
./db_load data/netrhost.gdbm data/netrhost.db.flat data/netrhost.db.new
```

F. Your ./Startmush should re-index all your txt files you originally made:

```
./Startmush
```

G. In your game type the following as an immortal or as #1.

1. Load in your mail flatfile:

```
wmail/load
```

2. Load in your autoregistration flatfile (if available):

```
@areg/load
```

3. Load in your hardcoded bbs flatfile (if available):

```
newsdb/load
```

9. You should be good to go on a current directory structure for Rhost. Enjoy!

13.3 Adding MySQL to RhostMUSH older than 3.9.5p2

MySQL is now native in RhostMUSH as of 3.9.5p2.

Warning: To autodetect it, YOU MUST HAVE mysql_config installed and running on your server. Without this, even if you have mysql db installed it won't be able to recognize the parameters you will need for it and will thus fail. Please check your linux distribution to see what packages are needed to install mysql_config.

Download the git repository to a seperate directory so that you can copy over the files that it requires you to.

Suggestion:

```
git clone https://github.com/RhostMUSH/trunk ~/tmprho
```

If you are patching UP from an older version, you need to update the following files:

1. update your src/Makefile to the one in the 3.9.5p2+ repo (cp ~/tmprho/Server/src/Makefile ~/Rhost/Server/src/Makefile)
2. update your bin/asksource.* files to the one in the 3.9.5p2+ repo (cp ~/tmprho/Server/bin/asksource.* ~/Rhost/Server/bin/)
3. append 'include rhost_mysql.conf' BEFORE the rhost_ingame.conf file and before the section that says 'define local aliases' toward the end of your netrhost.conf file. (edit your ~/Rhost/Server/game/netrhost.conf file)
4. copy the game/rhost_mysql.conf file from the 3.9.5p2+ repo (cp ~/tmprho/Server/game/rhost_mysql.conf ~/Rhost/Server/game/)
5. The following lines have to be REPLACED/CHANGED in local.c (toward the top): (you may edit this or copy the one from the other distro) (do either: edit ~/Rhost/Server/src/local.c) (or: cp ~/tmprho/Server/src/local.c ~/Rhost/Server/src/local.c)

Note: IF REPLACING/CHANGING local.c COPY BELOW

```
/* Called when the mush starts up, immediatly prior to the main game
 * loop being entered. By this point all databases are loaded and
 * all variables configured.
 */
#ifdef MYSQL_VERSION
   extern void local_mysql_init(void);
   extern int sql_shutdown(dbref player);
#endif

#ifdef SQLITE
   extern void local_sqlite_init(void);
#endif /* SQLITE */

void local_startup(void) {
#ifdef SQLITE
   local_sqlite_init();
#endif /* SQLITE */
#ifdef MYSQL_VERSION
   local_mysql_init();
#endif
   load_regexp_functions();
}

/* Called immediatly after the main game loop exits. At this point
 * all databases and variables are still configured
 */
void local_shutdown(void) {
#ifdef MYSQL_VERSION
   sql_shutdown(-1);
#endif
}
```

6. Issue ‘make clean’ then make confsource to rebuild using the latest builder script to build in the mysql changes.

CHAPTER

FOURTEEN

COMPARISON OF FLAGS

PENN/MUX Flag	RhostMUSH Flag	Commentary on Difference Between Flags
ABODE	ABODE	
BLIND	BLIND	
CHOWN_OK	CHOWN_OK	
DARK	DARK	
FREE	FREE	
GOING	GOING/BYEROOM	
HAVEN	HAVEN	
INHERIT	INHERIT	
JUMP_OK	JUMP_OK	
KEY	KEY	
LINK_OK	LINK_OK	
MONITOR	MONITOR	
NOSPOOF	NOSPOOF	
OPAQUE	OPAQUE	
QUIET	QUIET	
STICKY	STICKY	
TRACE	TRACE	
UNFINDABLE	UNFINDABLE	
VISUAL	VISUAL	
WIZARD	ROYALTY	
ANSI	ANSI/ANSICOLOR	
PARENT_OK	PARENT_OK	
ROYALTY	COUNCILOR/ARCHITECT	
AUDIBLE	AUDIBLE	
BOUNCE	BOUNCE	
CONNECTED	CONNECTED	
DESTROY_OK	DESTROY_OK	
ENTER_OK	ENTER_OK	
FIXED	NO_TEL	
UNINSPECTED	Not Available	Just a marker flag
HALTED	HALTED	
IMMORTAL	GUILDMASTER	You don’t want IMMORTAL
GAGGED	FUBAR	
CONSTANT	NO_MODIFY	
LIGHT	LIGHT	
MYOPIC	MYOPIC	
AUDITORIUM	AUDITORIUM	

continues on next page

Table 1 – continued from previous page

PENN/MUX Flag	RhostMUSH Flag	Commentary on Difference Between Flags
ZONE	Use @zone	
PUPPET	PUPPET	
TERSE	TERSE	
ROBOT	ROBOT	
SAFE	SAFE	
TRANSPARENT	TRANSPARENT	
SUSPECT	SUSPECT	
VERBOSE	VERBOSE	
STAFF	Not Available	Just a marker flag
SLAVE	SLAVE	
ORPHAN	Not Available	@lock/use the parent instead
CONTROL_OK	Not Available	Use @lock/ZoneWizLock
STOP	STOP	See also NOSTOP
COMMANDS	COMMANDS	
PRESENCE	Not Available	See: Reality Levels
NOBLEED	Not Needed	Rhost doesn't bleed ANSI
VACATION	Not Available	Just a marker flag
HEAD	Not Available	Just a marker flag
WATCHER	Not Available	@toggle MONITOR
HTML	Not Available	Rhost doesn't support Pueblo
REDIR_OK	Not Available	Rhost doesn't support @redirect
SPEECHMOD	Not Available	Rhost doesn't support @speechmod - use @icmd
MARKER0-MARKER9	MARKER0-MARKER9	

CHAPTER

FIFTEEN

COMPARISON OF POWERS

announce Can use the @wall command. Rhost Equiv: - FREE_WALL (@power)

boot Can use the @boot command. Rhost Equiv: BOOT (@power)

builder Can build, if the builder power is enabled. Rhost Equiv: ARCHITECT (flag)

chown_anything Can @chown anything to anyone. Rhost Equiv: CHOWN_OTHER (@power)

comm_all Like a wizard with respect to channels. Rhost has no hardcoded comsystem. You can tweek the softcode.

control_all Can modify any object in the database. (God-set only.) Rhost Equiv: TwinkLock (@lock)

expanded_who Sees the wizard WHO, and SESSION commands. Rhost Equiv: WIZ_WHO (@power)

find_unfindable Can locate unfindable people. see_hidden Can see hidden (DARK) players on WHO, etc. Rhost Equiv: WHO_UNFIND (@power)

free_money Unlimited money. Rhost Equiv: FREE (flag)

free_quota Unlimited quota. Rhost Equiv: FREE_QUOTA (@power)

guest Is this a guest character? Rhost Equiv: GUEST (flag)

halt Can @halt anything, and @halt/all. Rhost Equiv: HALT_QUEUE (@power) or HALT_QUEUE_ALL (@power)

hide Can set themselves DARK. Rhost Equiv: NOWHO (@power)

idle No idle timeout. Rhost Equiv: @timeout *player=-1

link_variable Can @link an exit to "variable". Rhost Equiv: Anyone can do this. VARIABLE (@toggle)

link_to_anything Can @link an exit to any (non-variable) destination. Rhost Equiv: @lock/link (@lock)

long_fingers Can get, look, whisper, etc from a distance. Rhost Equiv: LONG_FINGERS (@power)

no_destroy Cannot be @toad'ed. Rhost Equiv: INDESTRUCTABLE (flag)

open_anywhere Can @open an exit from any location. Rhost Equiv: @lock/open (@lock)

poll Can set the @poll. Rhost has nothing equivelant. Just softcode a +poll, or @hook it for permissions.

prog Can use @program on players other than themself. Rhost Equiv: PROG (@toggle)

search Can @search anyone. Rhost Equiv: SEARCH_ANY (@power)

see_all Can examine and see attributes like a wizard. Rhost Equiv: EXAMINE_FULL (@power) (and EXFULLWIZATTR (@toggle) for wiz only attribs)

see_queue Can @ps/all or @ps any player. Rhost Equiv: SEE_QUEUE (@power) or SEE_QUEUE_ALL (@power)

stat_any Can @stat any player. Rhost Equiv: STAT_ANY (@power)

steal_money Can give negative money. Rhost Equiv: STEAL (@power)

tel_anywhere Can teleport anywhere. Rhost Equiv: TEL_ANYWHERE (@power) or FULL_TEL (@power)

tel_anything Can teleport anything (includes tel_anywhere) Rhost Equiv: TEL_ANYTHING (@power)

unkillable Cannot be killed with the 'kill' command. Rhost Equiv: NOKILL (@power)

use_sql Can call the SQL() function. (God-set only.) Rhost Equiv: SQL is a 3rd party patch.

watch_logins Can set or reset the WATCHER flag on themselves. Rhost Equiv: MONITOR (@toggle)

CHAPTER

SIXTEEN

RHOSTMUSH INTERNAL HELP FILES

CHAPTER

SEVENTEEN

RHOSTMUSH INTERNAL WIZHELP FILES

CHAPTER

EIGHTEEN

HISTORICAL NON-DISCLOSURE AGREEMENT

The following NDA comes from the time when RhostMUSH was not publially available. It is preserved here for historical reasons. We are suckers for looking back at things. :) –Ambrosia

1) I agree, to not give out the code, in part or in full, in any form of medium, to anyone or anything not previously allowed by the developers.
2) I agree, to not let others look at the code, in part or in full, in any form of medium, to anyone or anything not previously allowed by the developers.
3) I am aware that any modifications I make to the code is _FULLY_ permitted, and that I do _NOT_ have to return said patches to the developers.

Note: The Rhost developers would like to see what was added, and possibly look at adding them to the main distro if we see others would like it (with full credits to you), but we belive once you have the code, you should be allowed to play with it fully as long as the first two rules are kept.

CHAPTER

NINETEEN

INDICES AND TABLES

- genindex
- search