

Table of Contents

Introduction.....	1
System Requirements.....	1
Installing.....	2
Java Webstart.....	2
Autopackage.....	2
Warp-In.....	3
Windows self-extracting installer.....	3
Zip files and tarballs.....	3
JarFile.....	3
Starting.....	3
Parts of JamochaMUD.....	3
The JamochaMUD Connector.....	3
The JamochaMUD Main Window.....	4
Using JamochaMUD.....	5
Resizing the Input / Output windows.....	5
Saving output from a MU* session.....	6
Disconnecting.....	6
Pausing output.....	6
Releasing paused output.....	6
Scrolling output from the MU*.....	6
Command History.....	6
Configuring JamochaMUD.....	7
Backing up your preferences.....	7
JamochaMUD Plug-ins.....	7
Enabling and Disabling plug-ins (post 2004-07-27).....	7
Enabling and Disabling plug-ins (older versions of JamochaMUD).....	8
Installing new plug-ins.....	8
Configuring plug-ins.....	8
Plug-ins that ship with JamochaMUD.....	8
How To Create “typing short-cuts”.....	10
Variables.....	10
Definitions.....	10
Pretty Fonts.....	12
TinyFugue Emulation.....	12
Quitting JamochaMUD.....	13
Troubleshooting Problems.....	13
Frequently Asked Questions.....	13
License.....	13

Introduction

JamochaMUD is an open source Java MUD/Muck client.

System Requirements

JamochaMUD should run on any system that is running Java version 1.1.x or higher. All parts of JamochaMUD are written entirely in Java so should not require anything that is not available in the normal Java environments. It is important to note that if you are running Java 1.2 or Java 1.1 with Swing

support, JamochaMUD will take advantage of the more advanced components which may make your mucking/mudding experience more enjoyable.

Please note that you will need a correctly installed Java environment for JamochaMUD to work.

Installing

Several different installer archives are available for JamochaMUD; it is recommended that you download and install the archive type that you feel most comfortable with.

Java Webstart

A part of many Java distributions, Java Webstart lets you download, install, and run Java software simply by clicking on a link on a web page. By clicking the JamochaMUD Webstart link on the JamochaMUD download page (trusting that Webstart is properly setup on your machine) you will see a dialogue box appear. It will caution you that JamochaMUD was signed by an “untrusted” source (well, *really!*) and that you can either stop the download, trust the application once, or always trust it. Unless you don't want to run JamochaMUD, you should choose “always trust”. With only trusting the application once, it will download and install, but will not operate properly.

The reason that you need to trust JamochaMUD is that the MU* client needs to read and write to your hard drive to save its settings, save logs, and then read those settings again later.

More than likely Java Webstart will give you the option to make a desktop icon for JamochaMUD. After installation, whenever you start JamochaMUD from this icon it will automatically check the website for new versions. If there is a new version it will download it first, otherwise it will use the copy of JamochaMUD that is already on your system.

This is probably one of the easiest ways to get going.

Autopackage

Autopackage archives are a type of “universal installer” for Linux. This JamochaMUD download is wrapped in a basic shell script. You should be able to install this archive by either clicking / double-clicking on the link (depending on your system setup) or executing the archive from the command-line (such as `./JamochaMUD-1.1.051111.x86.package`).

When the script runs, it first checks your system to see if you have the Autopackage tools already installed. If not, the script downloads and installs them for you, before proceeding with the installation of JamochaMUD. When the installation is complete Autopackage will create a menu icon for JamochaMUD and add the deinstall information to Autopackage (that may be accessed through the obscurely named “Manage 3rd Party Software” icon or via the command line).

This archive assumes that you have the Java executables in your PATH.

More information about Autopackage may be found at <http://autopackage.org/>

Warp-In

Warp-In archives are for use only with OS/2 and eComStation systems that already have Warp-In installed. Warp-In may be found at this link: <http://www.xworkplace.org>

Once you download a Warp-In archive the installation can be started simply by double-clicking on the archive's icon. The installer will walk you through the rest of the installation, automatically recognizing any previous Warp-In installs of JamochaMUD (and using the same directories unless you specify otherwise). When the Warp-In install has completed JamochaMUD icons will be created.

Windows self-extracting installer

By downloading and double-clicking on the Windows self-extracting installer version of JamochaMUD the installation process will begin. JamochaMUD icons will be created both on your desktop (if you choose) and under your "Programs" menu item.

Zip files and tarballs

Both zip files and tarballs are two no-frills compressed archives of JamochaMUD. These archives must be decompressed using your favourite decompression program (such as Winzip, unzip or tar, depending on the archive type). Using these types of archives will not create icons to start JamochaMUD nor setup the operating environment required to run JamochaMUD. The system's classpath must be updated to include the directory that contains the anecho directory, and then JamochaMUD can be launched by using: `java anecho.JamochaMUD.JMUD`

JarFile

A JarFile is another compressed type of archive, but unlike a Zip file or tarball does not need to be uncompressed to be used. Instead the jarfile contains additional files that indicate to the Java runtime while class needs to be executed and how the environment should be setup... thus, starting a JarFile is very simple. You can use a command such as `java -jar JamochaMUD.jar` to get started.

Starting

Most of the above installers provide the means to start JamochaMUD via icons. Unless you know what you are doing (and typing something like `java anecho.JamochaMUD.JMUD` at the command line means something to you), we recommend you use Webstart or an installer.

Parts of JamochaMUD

The JamochaMUD Connector

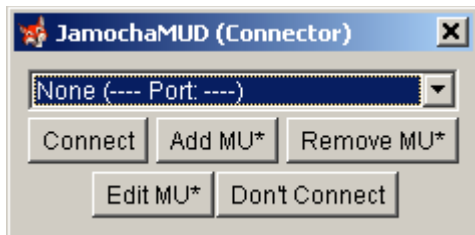


Illustration 1 The JamochaMUD connection window.

When JamochaMUD is first started, the Muck Connector appears. This dialogue allows the user to choose an existing MUD/Muck from a list and connect, edit a list item, add a new list item or simply start JamochaMUD without making a connection.

Connect To MU*

Choosing **Connect** will signal JamochaMUD to attempt a connect to the currently listed address, as well

as displaying the main JamochaMUD window if not already present.

Add a new MU* connection

Clicking *Add MU** brings up a new dialogue that allows you to enter a name, address, and port number for a new Muck/MUD connection. The name is entirely for your own reference, and should be used to describe this connection. The address and port are required for a successful connection.

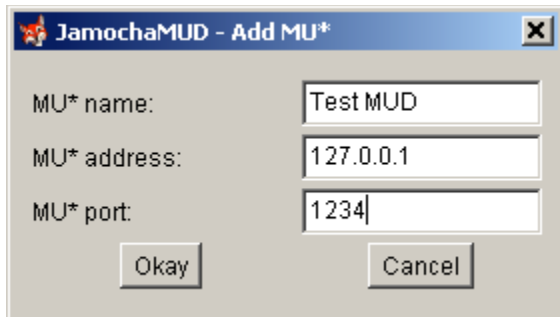


Illustration 2 The dialogue used to add or edit a MU*. This illustration needs to be updated to show the secure connection check-box.

Secure Connections

Newer versions of JamochaMUD now support secure connections via SSL (Secure Sockets Layer). If a MU* supports secure connections you may also enable this on the Add MU* / Edit MU* dialogue. Secure connections cannot be used unless the MU* you are connecting to also supports this feature.

A secure connection operates the same as a standard connect as far as the user is concerned. But instead of a plain telnet connection being used where plain data can be intercepted or monitored, the data is encrypted by JamochaMUD and only unencrypted when it reaches the destination server. Any of the data collected inbetween will be encrypted and unintelligible to anyone that gets their hands on it.

Edit an existing MU* connection

Choosing *Edit MU** brings up a dialogue like the one used to add a new connection, but will list the particulars of the MUD/Muck that is listed in the initial connector window. Here, changes may be made to the name, address and/or port. Choosing *Okay* saves these changes while choosing *Cancel* disregards the changes made.

Removing a MU* connection

Choosing *Remove MU** will delete the currently displayed entry. Please note that once an entry is deleted you cannot get it back without adding a new MU* with the previously deleted information.

Starting without connecting

Selecting the *Don't Connect* button will close the MU* Connector dialogue and bring up the main JamochaMUD window. If you wish to connect to a MU* once the main window has displayed, you may choose *Connect To MU** from the *Connections* menu.

The JamochaMUD Main Window

The main JamochaMUD window will look similar to the following image, and is divided into three areas from top to bottom: The menu bar, the output window and the input window.

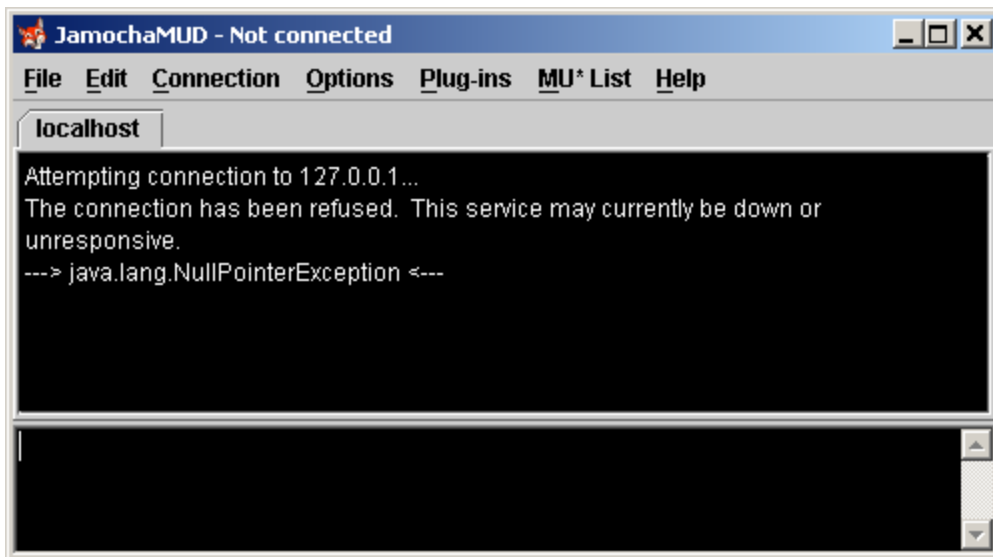


Illustration 3 The main JamochaMUD window. This features the frame title, main menu, MU* tab, output window and input window.

Menu Bar

The menu bar operates like the standard menu in other software, offering you commands to harness the raw power of JamochaMUD!! (The commands will be discussed in more detail below).

Output Window

The JamochaMUD output window will present you with any output coming from the connected MU* as well as any important notices from JamochaMUD itself (such as the error in this example). On machines running Swing/Java 1.2+ there will be a tab at the top of the output window displaying the name of the currently connected MU*. A tab will appear for each connection, allowing you to easily switch between active sections. On systems without Swing/Java 1.2+ the different connections are only accessible via the command short-cut or by using the *Connections* menu options.

Input Window

The Input Window is where any commands to be sent to the MU* can be typed. In addition to MU*-based commands, there are also special commands that JamochaMUD can also handle to make your mucking/mudding experience more enjoyable.

Right-clicking on the input window will present you with a pop-up menu listing the last 10 commands input into the MU*.

Using JamochaMUD

Resizing the Input / Output windows

The relative space of the Input and Output windows can be adjusted if you are using JamochaMUD's Swing interface (available only with Java 1.2 and higher). A “grippy” will appear dividing the input and output windows. By clicking the mouse and dragging the “grippy” the two areas can be resized.



Illustration 4 The horizontal grippy between the input and output window.

Saving output from a MU* session

By choosing the option “write log to file” found under the File menu you will be presented with a save file dialogue box. You can specify the name of the file you wish to save and any extension needed (having .txt on the end of your file name may be beneficial on Windows-based systems). Currently, JamochaMUD will only save the output as a plain text file, but future versions may support HTML output to preserve colours and formatting.

Disconnecting

There are three ways that you can initiate disconnecting from a MU* when using JamochaMUD.

- By closing JamochaMUD (either by clicking the close icon on the window frame or choose Quit from the File menu) the connection to your MU* is automatically terminated.
- If you have more than one MU* connect open, you may choose Close this view from the Connections menu.
- A connection may also be terminated by choose Disconnect from MU* from the Connections menu.

Pausing output

The output window of JamochaMUD may be paused at any time by single-clicking the mouse pointer on the output window. The title of the JamochaMUD frame will change to read “Output Paused (press any key to resume)”, such as in the image below.

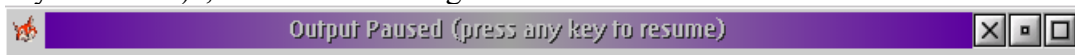


Illustration 5 An example of the frame title when output is paused.

JamochaMUD will continue to accept output from the MU* server, but will not show it until you released the pause.

Releasing paused output

To release paused output in JamochaMUD, simply type any key (no, not the “Any key”) and the pause will be released. The pause may also be released by clicking the mouse in the input window or, if TinyFugue key emulation is enabled, by using the CTRL+Q key combination.

Once the output pause has been released, the title on the JamochaMUD frame will return to it's previous reading.

Scrolling output from the MU*

In addition to using the scrollbars found on the MU* output window of JamochaMUD, you may also jump forward and backwards a page at a time by using the key combinations of Shift+Page Up (to scroll up) and Shift+Page Down (to scroll down).

Command History

JamochaMUD can keep track of the commands that a user types, in the event that they need to be

recalled. The command history may be accessed by right-clicking on the input window in JamochaMUD or by using the key combination of CTRL + N (if TinyFugue emulation is enabled). The last typed (previous) command may also be recalled using the CTRL + P combination (if TinyFugue emulation is enabled).

The number of entries contained in by your command history may be changed by navigating to *Options -> Configure JamochaMUD -> Command History*.

Configuring JamochaMUD

Backing up your preferences

Starting with the 2006-02-05 release of JamochaMUD settings and preferences can easily be backed up and restored in a platform independent way. This allows a user to back-up a configuration, for example, on Linux and restore it in Windows. This back-up procedure will back-up all JamochaMUD settings and settings for Plug-ins that are coding to the standards outline in JamochaMUD.

Saving

To back-up your settings choose *File -> Export JamochaMUD settings*. You will be asked for a name for the back-up file (the extension of .jsa is automatically added for a JamochaMUD Settings Archive) and a location to save it. Once the setting archive has been created, you may continue using JamochaMUD normally.

Restoring

To restore settings from a JamochaMUD Settings Archive navigate to *File -> Import JamochaMUD settings*. You will then see a file dialogue that will ask you to locate your .jsa file. When you find this file the settings will be restored. JamochaMUD will then shut itself down. When the user restarts JamochaMUD the restored settings will be used.

Customised output colours

JamochaMUD allows you to customise the output colours presented by different MU*s. When accessing this dialogue from *Options -> Configure JamochaMUD -> Output Colours* you will be presented with a grid of 16 colours. The left column represents the “low colour” range and the right column represents the “high colour” range.

Double-clicking on any of the cells in the table will allow you to modify the colour.

The custom colours only over-ride ANSI colours that are set by MU*s, and will not affect your regular foreground and background colours.

Standard telnet colours may be restored at any time by clicking the “Reset Colours” button.

JamochaMUD Plug-ins

All information related to JamochaMUD plug-ins is located under the Plug-ins menu item on the main JamochaMUD window.

Enabling and Disabling plug-ins (post 2004-07-27)

In more recent versions of JamochaMUD, the handling of enabling and disabling plug-ins has now been moved from a separate dialogue box to be handled directly in the plug-ins menu itself. On Swing-based Java Virtual Machines a green or red globe will appear beside the plug-ins' name. Green indicates that

the plug-in is enabled, while red indicates that the plug-in is not currently enabled. A sub-menu on each of the plug-ins' names will give you the menu options of viewing the plug-in's properties, enabling or disabling the plug-in and getting a description of the plug-in.

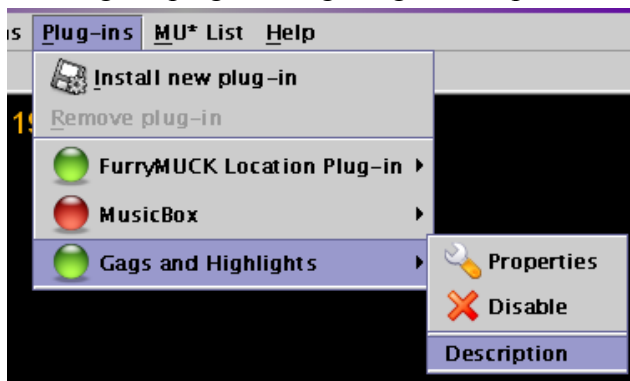


Illustration 6 Displaying the properties menu for an installed plug-in.

Properties are user-definable options of the plug-in. If a given plug-in does not have any user options to it the Properties menu item will be disabled (greyed out).

If the plug-in is already enabled, a “**Disable**” option will appear on the sub-menu. Should the plug-in be disabled, an “**Enable**” option will appear instead.

The **Description** menu item will present you with a dialogue box giving a brief description of the functions the given plug-in performs.

Enabling and Disabling plug-ins (older versions of JamochaMUD)

versions of JamochaMUD)

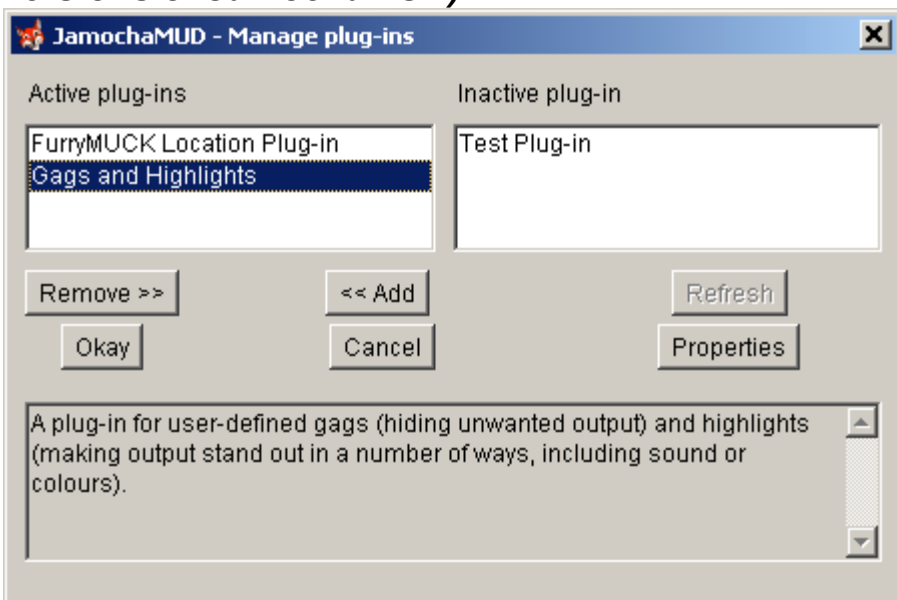


Illustration 7 The old way to access plug-in properties.

In previous versions of JamochaMUD plug-ins had to be enable or disabled via the “Manage Plug-ins” dialogue box.

Plug-ins can be individually enabled and disabled by going to the plug-in configuration dialogue, located under the Plug-ins -> Configure Plug-ins menu item.

In this dialogue you will see two lists, one containing active plug-ins and one containing inactive plug-ins. To activate a plug-in, click on the name of the plug-in in the inactive list and then click

then “Add” button. To deactivate a plug-in, click on the name of the plug-in in the active list and then click the “Remove” button.

Installing new plug-ins

To install a new plug-in, navigate to the Plug-ins -> Install plug-in menu item. You will be presented with a file dialogue where you can navigate to the location that you have downloaded the plug-in archive to. The plug-in archive should have the extension of .jpa (JamochoMUD Plug-in Archive).

Selected the archive and confirm your choice. JamochaMUD will proceed with installing the plug-in in the proper location. You may need to restart JamochaMUD to take advantage of the new plug-in.

Configuring plug-ins

Plug-ins may be configured in two different ways. Either choose the plug-in's name from under the Plug-

ins menu on the main window, or choose Plug-ins -> Configure Plug-ins menu item. Not all plug-ins have user changeable configurations.

Plug-ins that ship with JamochaMUD

JamochaMUD comes with a few plug-ins installed already (though they may be disabled when you first run JamochaMUD). The details of these plug-ins will be described below.

FurryMuck location plug-in

Most likely this plug-in will only be useful if used on the FurryMUCK server, though it may also work for other mucks that run Fuzzball servers. Being mostly benign, this plug-in simply watches output from the MU* for [fmlc] tags, which contain the names of different locations on the muck. When you enter a new location this plug-in will parse that tag and place the name of the room/location as the title of JamochaMUD's window.

There are no settings or options for this plug-in.

MusicBox plug-in

This idea was blatantly stolen from the LiveJournal client DeepestSender (<http://deepestsender.mozdev.org/>). This plug-in can be setup to look for a music information file that can be created by most "mp3" playing software.

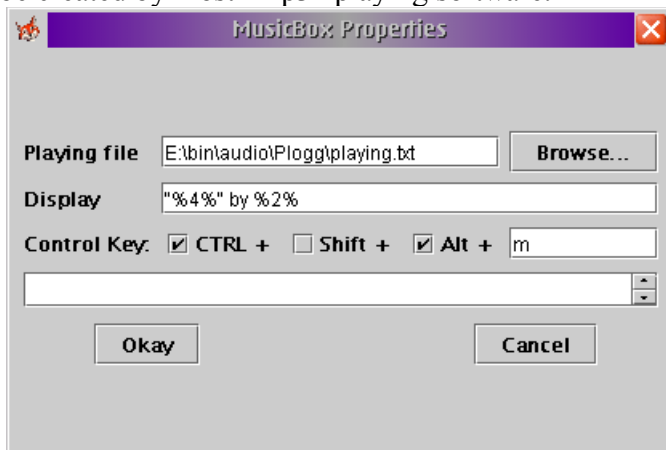


Illustration 8 The configuration dialogue for the MusicBox plug-in.

In the example shown at left, we have selected the "Playing file" which contains the song information written out by the mp3 player. In this case it is called "playing.txt" and located in the e:\bin\audio\Plogg directory. Simple enough, huh?

The "Display" line tells the plug-in how to paste the music information into your text input window. Any number surrounded by percentage signs (such as %4%) tell MusicBox to insert the information from that line into the position of the number and percentage signs. In this case, if our playing file contents looked like this:

Now playing:

Nick Cave & The Bad Seeds

No More Shall We Part

God Is In The House

2001

and having "%4%" by %2% as our display will result in "God Is In The House" by Nick Cave & The Bad Seeds being pasted into the text input area of JamochaMUD.

Of course, we need a way to activate MusicBox, and that is where the Control Key comes in. You can tell MusicBox that you want a combination of CTRL, Shift, and Alt along with an alpha-numeric key to paste the music information into JamochaMUD. In this example, hitting CTRL+ALT+m will have the desired results!

Gags & Highlights

Most likely one of the most useful plug-ins that comes with JamochaMUD, the Gags & Highlights plug-in allows you to customise the look of the output you receive from the MU* server. You can use this to colour the output, emphasis it, or even block it if you so desire!

This plug-in works on the basis of “rules”, which may seem strange at first, but once you get the hang of it you should be able to create some very useful combinations.

How To Create “typing short-cuts”

One of the interesting features that JamochaMUD has is the ability to create definitions and hold variables that can be setup by the user. This is basically a way of creating short-cuts to frequently typed items with very little effort.

Variables

A variable can be thought of as the label for a container. You can fill this “container” with whatever text you would like and recall the text at any time by just giving JamochaMUD the name of the label of the container (the variable name).

Say we had a user named *Ralphamungous* that we talked to quite regularly (or perhaps talked about quite regularly). It's quite a bit of typing to repeat *Ralphamungous* all the time, especially if you are in an instance where spelling counts! To ease the burden you can instead use a variable.

```
/set ral=Ralphamungous
```

By typing in the `/set` command we've told JamochaMUD that we want to make a variable named *ral* and associate *Ralphamungous* with that variable. Now, whenever we want to have the name *Ralphamungous* appear we only need to tell JamochaMUD to use the variable. “But how”, you ask? Very simple... just wrap our variable name in a brackets and then put a dollar sign in front. This special formatting lets JamochaMUD know it has to do a replacement (otherwise it would just send everything straight to the MU* as-is). An example of this would be:

```
“Hi, ${ral}, how are you doing?”
```

JamochaMUD will then do the variable substitution and send to the MU*:

```
“Hi, Ralphamungous, how are you doing?”
```

Now, say the character *Ralphamungous* becomes no more, so we don't need to have the *ral* variable hanging around any more. We can simply unset the variable by telling JamochaMUD:

```
/unset ral
```

JamochaMUD should give you a little confirmation that the variable has now been unset and we can move on with our lives! But, what if you've been using a lot of variables and you can't really keep track of what you have anymore? Simply type:

```
/set
```

in the input box and JamochaMUD will nicely list out all your variables and what their associated settings are.

Definitions

Definitions operate in a way that is similar to variables, in that you can give a definition a label and then have it associated with a larger amount of text. But what definitions offer that Variables don't is also a certain amount of “logic” and flexibility. Variables tend to be on a “this equals that” basis, and will

always stay the same unless you change the value of the variable.

The best way to explain a definition is to give you an example. First, we'll set a definition:

```
/def greet=:waves to %1 in a crazy fashion.
```

Okay, that looks a fair bit like how we set a variable, except that we use the */def* command to tell JamochaMUD it is a definition, instead of the */set* command we use for setting up variables. We've called our definition *greet* and told it that when we call *greet* it should send out *:waves to %1 in a crazy fashion*.

But what is this *%1* in the definition? This is where the magic comes in. But first, we'll show you how we'd call a definition.

Unlike a variable where we have to go */\${ral}* to get it to function, for a definition to be used, we just have to start our input with:

```
/greet
```

Unfortunately, if we tried the command with just */greet*, we'll get:

```
waves to in a crazy fashion.
```

Sent to the MU*. What's going on? It's that magical little *%1*, which does our substitution for us. The *%1* tells JamochaMUD to insert the "first word" that it receives after the our definition name. For example, we can type:

```
/greet Ralphamungous
```

and JamochaMUD will translate this to:

```
:waves to Ralphamungous in a crazy fashion.
```

Okay, that seems like it'd be a nice way to personalize your greeting without too much work, but there is that nasty-long name again. So why don't we feed a variable to our definition? Working from the previous variable example, we could type in our definition as so:

```
/greet ${ral}
```

JamochaMUD will helpfully expand the variable first into *Ralphamungous* and then the definition will take it from there. In the end we'll come out with:

```
:waves to Ralphamungous in a crazy fashion.
```

just like before, but with even less typing!

Now, as you may have already figured out by our example that uses *%1* as a place-holder, you can have multiple place-holders in your definition. We could change our definition to be:

```
/def greet=:waves to %1 in a %2 fashion.
```

and now if we feed our greeting:

```
/greet ${ral} lazy
```

JamochaMUD will send out:

```
:waves to Ralphamungous in a lazy fashion.
```

All fairly straight forward, right? But what if we don't want to always type in what kind of fashion we wave in, but we don't want to create two different definitions just for a slight variation on our wording? Ah, yes... we can have default place holders, too! That means if we use our definition but don't give it everything it is looking for, it will use the default information instead. To set a default we can simply go:

/def greet=:waves to %1 in a %{2-crazy} fashion.

Our new bit of information here, the *{2-crazy}*, looks a bit like a cross between our definition and a variable, and indeed it is somewhat similar. The *{2-crazy}* tells JamochaMUD to use the second placeholder the user gives, but if the user neglects to give us one, we'll just use *crazy* instead! Examples of this would be:

/greet Jeff

which will send out to the MU*

:waves to Jeff in a crazy fashion.

But using the same definition and supplying two placeholders like:

/greet Jeff wacky

will be translated into:

:waves to Jeff in a wacky fashion.

And, as we've seen in a previous example, our placeholders and their defaults can also be passed any variable that we've setup previous, affording us quite a bit of flexibility in what we're working with!

At any time you can remove a definition simply by typing */undef* followed by the definition's name:

/undef greet

And also, much like the variables, you can view all your definitions by simply typing:

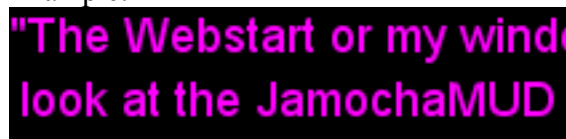
/def

in the JamochaMUD input window and pressing ENTER. (Or RETURN, depending on your preference and/or hardware).

Pretty Fonts

You may notice the first time you start up JamochaMUD that the fonts in the output window are not all that aesthetically pleasing. If you are running JamochaMUD on a platform that supports Swing (and/or Java 1.2+) you can do something about it! Under the Options -> Display Options menu you will find a check-box for Anti-aliased fonts. By turning anti-aliasing on JamochaMUD will “smooth” the edges of the font to make to more rounded and easier to read. You may find that you have to increase the size of your fonts if you choose this option, as the anti-aliasing can sometimes make the characters appear more dim than without.

Example:



"The Webstart or my wind
look at the JamochaMUD

*Illustration 9*Text with anti-aliasing enabled



"The Webstart or my wind
look at the JamochaMUD

*Illustration 10*Text with anti-aliasing disabled

TinyFugue Emulation

What is TinyFugue and why would I want to emulate it? TinyFugue is one of the more “traditional” unix-based (and powerful) MU* clients around. By itself, it operates solely in text mode, but comes with some wonderful features to make life easier... and part of what JamochaMUD has borrowed are its keyboard commands for editing text in your input window. By turning on TinyFugue emulation JamochaMUD maps these keys combinations to their following functions:

TinyFugue Editor Commands

CTRL+B -	jump left to beginning of word
CTRL+F	jump to end of word
CTRL+K	delete from the cursor to end of line
CTRL+N	show command history
CTRL+P	recall previous command
CTRL+Q	un-suspend output
CTRL+S	Suspend output
CTRL+T	Transpose characters
CTRL+U	Erase entire line at cursor's position

Quitting JamochaMUD

To quit JamochaMUD you may go to the File menu and choose Quit from the options available there. In addition, some operating systems allow you to click a “close” icon on the edge of the frame. Please note that when you close JamochaMUD you will not be prompted to save any output from your current MU* sessions.

Troubleshooting Problems

Should you run into problems while running JamochaMUD, please check the JamochaMUD Mantis bug-tracker located at <http://www.anecho.mb.ca/mantis>

Here, you can login anonymously to view known, existing and closed issues with JamochaMUD as well as using the search feature to look for other problems that may be the same as the one you are experiencing. Should you find a bug that is the same as what you are already experiencing you can choose the “Monitor Bug” option (when viewing the bug details page).

If the problem you're having is not listed, feel free to start a new bug report. After all, if a problem isn't reported, nobody else may ever know to fix it!

Please note: You may have to login to an account (and create one if you do not already have one) to report or monitor a bug.

Frequently Asked Questions

License

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to

using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

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

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

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

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

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

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

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

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

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

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

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

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

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

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

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

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

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

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

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

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

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

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

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

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

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

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

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

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

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

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

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

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

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

NO WARRANTY

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

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

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy <name of author>
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
`Gnomovision' (which makes passes at compilers) written by James Hacker.
```

```
<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

