The Palace™

Wizard’s Guide

The Palace Wizard’s Guide

November, 1998

The contents of this document are for informational use only, and the contents are subject to change without notice. Electric Communities assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Printed in the USA.

Contents

Preface

Welcome to *The Palace Wizards Guide;* the handbook for Palace wizards and gods. Whether you're a newly-appointed wizard in search of a place to begin your studies, looking for a refresher on your command syntax, you've come to the right place. Within these pages will be exposed the arcane mysteries of The Palace software and its inner workings: from the secrets of the fabled *wizard menu* to the powers of *authoring mode*; from the commands for creating rooms and interactive objects to the special site-wide messages received only by the initiated. Within these pages, all will be revealed...

This document is designed to assist wizards and gods in the performance of their palatial duties; from design and remote site maintenance to chat moderating and security.

Sections of this manual

This manual has the following:

- *Wizards and Gods*

  This section explains the roles and powers of these special Palace users, and examines the hierarchy of Palace society in general.

- *The Wizard's Interface*

  This section explains the advanced functions of The Palace client; most notably the Wizard Menu, which deals with the creation and online "editing" of your Palace site.

- *Wizard and god commands*

  This section describes the syntax and effect of all special commands available to both wizards and gods.

- *The Good Wizard*

  This section provides a suggested set of social guidelines for Palace authorities, many of which have been discovered "the hard way" (i.e., by trial-and-error). These "rules" are by no means mandatory — they are included here only for consideration as you establish the social order of your own Palace.

- *Palace Design Tips & Techniques*

  This section examines the design process, providing examples to illustrate various popular Palace effects. This section also describes some of the limitations of the Palace system (synchronization, bandwidth, memory, script size, stack space, and so on), and suggests workarounds for your more ambitious effects.

- Appendix A: "The ~club Command"

This section describes use of the ~club command, which allows The Palace software to be run safely and securely in public areas.

- *Appendix B: Command Summary*

  This section briefly describes the syntax and effect of all special commands available for user classes (i.e., guests, members, wizards and gods). It is intended for use as a "cheatsheet" while online.

Manual conventions

This manual has the following conventions:

- Code samples and text you enter are set in `Courier`, whether standing alone or embedded in the text.
- Technical terms appear in ***boldface italic*** in the text in their first appearance, usually accompanied by a definition.
- Graphical User Interface (GUI) elements, such as menu items and buttons, appear in **boldface**.
- References to other books, chapters, or sections are in *Italic*.

NOTE – Notes to the user look like this.

Because this document includes many references to command words and literal strings, *The Palace Canonical Notation* (i.e. the officially authorized method of code formatting) is used throughout. This is a set of conventions used to format code so the text is of maximum use from a technical point of view. Although many of these conventions are not strictly necessary from the standpoint of the code itself, canonical notation is recommended for those who wish to share scripts, as it increases their readability and ease of use.

The conventions of The Palace Canonical Notation are as follows:

- All code is shown in a `monospaced` font; commands and keywords are in `UPPERCASE`. This is not necessary as far as the parser is concerned, but it helps the code stand out from the data. For example:

  ```
  "Hello world" USERNAME & SAY
  ```

- Placeholders for actual values are written in *`italics`*. Placeholders for string values always appear in double quotes (`" "`). These double quotes must remain in the code; do not remove them. For example:

  ```
  "filename" MIDIPLAY
  ```

- All *Symbols* (variable names) begin with lowercase letters, although they may possess median capitals or underscores. Again; this is not necessary, but it makes your variables a little easier to find. Optional variables are enclosed in angle brackets (< >). For example:

```
myNewVariable
another_variable
`banip <minutes> ip
```

- Indentation indicates supporting code. All text supporting any ROOM, DOOR, SPOT, SCRIPT or ON (Event Handler) keyword is indented, and the indent is decreased when the supporting code ends (i.e., after any ENDROOM, ENDDOOR, ENDSPOT or ENDSCRIPT keyword, and on the closing curly bracket of an Atom List or Event Handler). This lets you tell at a glance where the code for each section starts and stops. For example:

```
ROOM
   ID 23
   NAME "The Pyramid"
   SPOT
      ID 1
      NAME "The Eye"
      SCRIPT
ON SIGNON {
   {
      "yes" SOUND
      1 ME SETSPOTSTATE
      {
         "@250,200 Hello " USERNAME & ", and welcome!" & LOCALMSG
      } USERNAME "Mister Wizard" NOT IF
   } 120 ALARMEXEC
}
      ENDSCRIPT
   ENDSPOT
ENDROOM
```

Basic Room Structure

Other sources of information

Besides this *Wizard's Guide*, the Palace provides other online manuals for administering your server, using the Palace client, and mastering Iptscrae. These manuals are all available from the Palace website in the Support section.

- *Personal Server Guides*. These manuals tells you how to obtain, install, setup, and run your own Palace personal server. Both a Windows 95/NT and Macintosh version are available.

- *Commercial Server Guides*. These manualsl tells you how to obtain, install, setup, and run the Palace commercial servers. These are servers intended for hosting many users. In addition to the features of the Personal servers, commercial servers also offer advertising, server-defined avatars, and the Palace Moderator, which is an auditorium-hosting system. Both a Windows NT and UNIX version are available.
- *User's Guide*. This manual describes how to obtain, setup, and use the Palace client software. You will need the Palace client to perform wizard and god actions for your site. Both a Windows 95/NT and Macintosh version are available.
- *Iptscrae Language Guide*. This manual is a programming and language reference guide for the Palace Iptscrae language.

In addition to the Palace online manuals, check out these pages in the Palace website for support, information, and online documentation:

Palace Home Page
http://www.thepalace.com

PalaceSpace (the page for Palace sites, including the Palace Directory)
http://www.palacespace.com/

Palace FAQ
http://www.palacespace.com/discover/faq/index.cgi

Palace Downloads (software & Assets)
http://www.thepalace.com/downloads/index.html

Palace Documents (Manuals and FAQs)
http://www.thepalace.com/support/manuals/index.html

Palace Tech Support
http://www.thepalace.com/support/

1 Wizards and Gods

This section describes wizards and gods:

- *Palace wizards* on page 9
- *Palace gods* on page 10
- *The Palace user hierarchy* on page 10
- *Personal styles* on page 11

Like most text-based chat systems and Internet MUDs (Multi-User Dimensions), the Palace software allows you to assign special powers to a select group of trusted users, making them “virtual authority figures” on your site. In Palace parlance, these individuals are known as ***wizards*** and ***gods***.

Palace wizards

Wizards are members who have been entrusted with special powers and responsibilities on one or another Palace server (including their own). They have the ability to create and destroy rooms, edit scripts, disconnect unruly users, and generally manage the place. The wizard's role is an interesting mixture of "host" and "security officer." Some lean more toward the *design* angle of the job, concentrating on creating new rooms and room-level scripts, while others prefer the *social* angle, and spend their time hanging out in The Palace, keeping things moving, interesting, and within community standards. Although the guidelines they’ll follow are up to the local Palace owner to establish, it is important to remember that all wizards are pretty powerful folks, who could potentially do as much harm to your site as good. For this basic reason, wizard status is not to be taken lightly.

All wizards may enter ***wizard mode*** using The Palace client. While in wizard mode, all wizards can:

- Edit scripts on the server

- Enter hidden, locked and private rooms
- Create and destroy rooms, doors and spots
- Change room settings, pictures and permissions
- Broadcast messages throughout the entire Palace Site
- Communicate secretly with other wizards

In addition to the functions provided via The Palace client interface, all wizards have access to a set of special commands known (not surprisingly) as "wizard commands." These commands and their use are also described in this document.

Palace gods

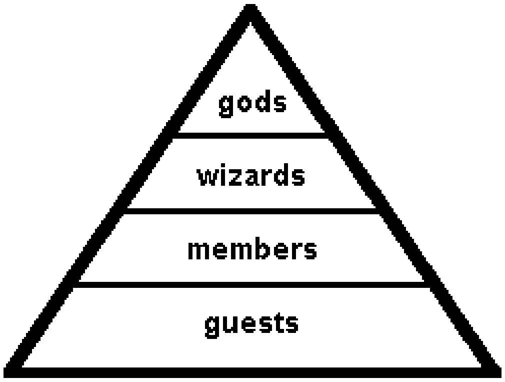
Above the level of wizards are the gods of your Palace. Gods possess all the powers of wizards, plus an additional set of ultra-powerful god commands. Besides possessing the ability to permanently ban users from the site, gods are able to remotely change the site’s passwords and permissions (maximum number of users allowed, guests allowed, killing allowed, etc.), or even bring the Server down via the client interface. god commands are so powerful, and their potential for misuse is so great, that it is extremely rare to encounter a god who is not also the owner of the site in question.

In character and function, Palace gods tend to be similar to the *design wizards* mentioned above – they are primarily concerned with creating, maintaining, running and evolving their Palace sites. Like wizards, Palace gods also possess a special set of commands that are covered in this document.

NOTE – The person who establishes the server and sets up the Server Preferences is the first default god of that Palace site. In setting up the site, this person will need to create the wizard and god passwords; these passwords will then be required for anyone (including themselves) to become wizards or gods.

The Palace user hierarchy

Among the users of any given Palace there are four User Classes which are defined by the capabilities granted them (either by the software itself or by the local Palace owners). These user classes effectively form a four-tiered model of Palace citizenry.



The Hierarchy of Palace Users

From the bottom up:

- ***Guests*** are generally new users of The Palace software. Guests possess only a limited set of user capabilities; they are unable to change their names or avatars, they may not manipulate props or use the painting tools, and they lack other advanced capabilities as well. All guests are called “Guest,” followed by a currently-unique ID number (e.g., “Guest 23”). This number is set by the server upon login to any Palace site, and cannot be changed directly.
- ***Members*** are either registered users of The Palace software, or guests who are using the limited "Trial Membership" time granted (at specified sites only) by the most current version of The Palace client. Members possess the full set of user capabilities as described in *The Palace Users Guide.*
- ***Wizards*** are members who have been given special powers by Palace gods. Most people become wizards by impressing the gods of a given Palace with their responsibility and maturity; others by their mastery of ***Iptscrae***. Wizards have the ability to enter ***authoring mode,*** allowing them to change the properties of all rooms, doors and spots in the Palace where their rank was granted (the local *wizard password* is determined by the local Palace gods, so a password for one site will not work at another).
- ***Gods*** are members who possess the ultimate powers in the Palace community, most notably the power to permanently ban other users from the site. The god(s) of a Palace site are generally the original creators of the site, or have been granted their god status by the site’s original creators. Like the wizard password, the *god password* is effective only in the Palace where it was granted.

Personal styles

In some ways, wizard duty is like paramilitary or emergency response team training: it provides a firm set of rules for progressing the goals of the site and defending the local citizenry from attack or disaster. Obviously, in the event of a "situation" (e.g., abusive guests, indecent props, serious scripting errors or other emergencies), all wizards on your staff should be familiar with your site's guidelines and recommended procedures. Their responses to emergency situations should always be consistent with your site's policies: they should all deal with similar situations in similar ways. They should also be able to handle these operations effectively as a team. Provided all wizards follow the site guidelines and stay in close contact, they can always shift roles to "cover each other" as the situation requires.

Most of the time, however, it's perfectly all right for your wizards to loosen up a bit, kick off their pointy shoes and let down their long white hair. After all, wizards are people too! Besides; paranoia isn't good for tourism.

There are many types of wizard; too many to enumerate here, since each individual (hopefully) brings his or her own best qualities to the job. People with advanced knowledge of Iptscrae will gravitate naturally toward script creation, those with people-skills will tend to take on "host" roles, and others will discover a real talent for online security procedures. In general, wizards are free to pursue their online interests like any other Palace user; they can chat, animate, play games, and do whatever everyone else does at your site. In fact, a casual visitor might not realize there are any wizards present at all!

This freedom to define your own role as a wizard (within reason) is one of the things that makes Palace sites so unique: there's always room for personal style. But above the level of personality, it can still be said that most wizards fall into one of two broad categories: they are either *design wizards* or *social wizards*. (It is very rare to find someone who truly excels in both of these categories.) It turns out that these two categories can be further divided into two, yielding four basic wizard styles: *Artists, Programmers*, *Hosts* and *Security*. Although these terms are used primarily for description's sake, they are useful in understanding the variety of roles that wizards can play. The following section describes the differences between these four basic styles, and why you should care.

Design wizards

Design wizards like to focus on the "physical" and "mechanical" aspects of the site; they tend to be graphic artists or computer programmers in "real life," and are therefore attracted to those aspects of The Palace which allow them to flex their artistic and intellectual muscles. For these types, The Palace is a huge blank slate waiting for them to create a masterpiece on it. Although their innate talents make them important members of the site team (at least during the initial development phase), they are generally more interested in creating new things than with hanging around looking at old ones. Depending on the nature and complexity of your site, it may or may not be necessary to retain these individuals after your Palace opens to the public: occasional maintenance or regularly-scheduled "upgrades" may be the best route to take.

Design wizards can be divided into *Artists* and *Programmers*. It is extremely rare to find both of these skills in the same person, since each is a highly specialized occupation in its own right.

The artist

The *artist* type will enjoy making props and room backgrounds of all sorts, and there's always room for another room, right? High-quality original art is certainly a desirable feature for any Palace site; therefore this type is usually a valuable addition to the site team, especially during the design (pre-launch) phase.

If you're this type of wizard (or if you have one on your staff), you might want to consider reading up on those Iptscrae commands that deal with *Spots*, in the section *Authoring* on page 57 (for more information see the *Spot Commands* section of the *Iptscrae Language Guide*). This "minimum dose" of the programming language will allow you to create more complex visual effects and animations, without getting too deep into the technical aspects. In fact, although fixing things permanently into place requires a small amount of scripting, new graphics can be added, removed, manipulated and frame-flipped without ever opening the *Scripting window*, so you can be sure to get your code right the first time. Choice projects for this type include props, prop dispensers, automated dressing rooms, animated avatars and various types of spot animation.

The programmer

The *programmer* is another unique type that should be encouraged to do what he or she does best: *code*. The programmer mindset is always looking for tricky ways to do more and more complicated things, and this generally works fine if they're given the time and space they need to solve the problem. But there's a cost: this need for mental isolation, almost by definition, makes social activity somewhat less important for this type. That means they're generally *not* the people to call when there's an emotional scene or a touchy diplomatic situation.

Since they are always attempting to "do the impossible," programmers — especially new ones — might sometimes end up creating effects that are unintended, buggy, or disruptive to the flow of the site. Coding is often done in an iterative process, moving back and forth between the server and the text editor, and the script tends to pass through a lot of "wrong" stages before it reaches the "right" one. Because of this, new room scripts should be created and tested offline, or during non-public hours. Any Iptscrae wizard worthy of the name should own a copy of *The Palace Personal Server*, which should be used locally for testing script revisions without affecting the public server. Once the room is finished, tested and approved by the Palace gods, it can be inserted into the server script as a whole.

Although Iptscrae is nowhere near as vast and arcane as many programming languages, its peculiar syntax and use of reverse polish notation makes the true Iptscrae wizard a pretty rare breed. If you're one of these, you're a member of a select group in Palace society, and you know it. Unfortunately this can often go to your head; programmers are notorious for taking control of projects, feeling that only they can "get the job done." As true as this might be in some situations, it's usually bad for the integrity of the team. In addition, due to your intelligence and technical knowledge, you can often expect others to understand more than they do. To be a truly effective wizard, you should try to temper your expectations, both of yourself and others. Remember that running a successful Palace site is a team effort; if you support others in their weaker areas, they'll support you in yours.

Social wizards

Social wizards are just what they sound like: they are chiefly concerned with human-to-human interaction, either via direct communication or Internet-based networking tools. This type prefers to hang out on the site in real-time, focusing their energies on dealing directly with people and events; they are the people your users will come to recognize as "wizards" when they visit your site. Accordingly, while Social wizards are not vital personnel during the design and construction of your Palace, it's advisable to have at least one of them around whenever your site goes online.

Like Design wizards, Social wizards fall into two main groups: they are usually either *Hosts* or *Security*. Unlike the two categories described above, these styles may often be found in the same person.

The host

Hosts are the most visible wizards on a site; they are the party-throwers, the event planners, and the conversation-starters of the group. A good Palace host is much like a good party host in real life: they should have a natural way with people; they should be understanding and tolerant of alternative views, and have a good sense of humor. They should be helpful and considerate, especially with new members and young users. They should also have a very clear understanding of the goals of the site as determined by the Palace god(s), since the hosts are the ones who will be around most often to ensure that these goals are being served.

People who would make good hosts are fairly easy to spot, since even before they are officially recognized, they tend to spend a lot of time explaining the software to people and helping others find their way around. One way to determine who would be a good host is to ask your users to vote for the member "most likely to be a good wizard." Since hosts are the most visible and socially-oriented of the four basic wizard types, the majority of users will identify "good wizards" with good "host" qualities.

If you're this type of person, you're a natural choice for wizardship. You probably spend a lot of time helping people on The Palace already; having a title would just make it "official." To stay on top of your game, you should be familiar with the section *Wizard commands* on page 29 *and Message shortcuts* on page 39. You might also want to check out *The Iptscrae Language Guide*; especially the section on *Cyborg Commands*.

The security officer

Viewed by rambunctious users as the host's "evil twin," the *security officer* type is a natural "guardian" who doesn't mind stepping in and getting directly involved when things get ugly or confrontational. They should be responsible, practical, assertive when necessary, and fast on their "virtual feet." Like Security Officers in real life, this type knows where the big guns are, and isn't afraid to use them. Unfortunately (and again, just like in real life), they may tend to perceive situations as being more dangerous than they really are, if only to justify bringing out their coolest defensive weaponry. Still, all other things being equal, they are probably the single most important wizard type to have around once your site goes public — just in case.

If the above paragraph describes your wizarding style, you need to do more than *read* some documents — you need to *train* with them. You should be familiar with all the *Wizard commands* on page 29 and *Message shortcuts* on page 39, especially the "targetable" ones. Whether or not you're already familiar with these commands, you might want to get a few of your friends online once in a while and use them as targets (set the death penalty for 2 minutes so they can come back quickly). The goal of this exercise is simply to sharpen your response time, test out the various commands and see how to use them to best suit the needs of your site. You should also download *The Iptscrae Language Guide* and get familiar with editing the `Cyborg.IPT` file: your job will be a lot easier with a few homemade Iptscrae tools in your security toolkit.

2 The Wizard's Interface

This section describes the advanced authoring features accessible by wizards and gods via the Palace software interface:

- *Becoming a wizard* on page 17
- *Becoming a god* on page 18
- *Using the Wizard menu* on page 19

Becoming a wizard

To access the authoring features of the Palace software you need to be a wizard, which means two things:

- You need to have access to The Palace client software (either the Windows or Macintosh version), since wizard commands are sent exclusively through The Palace software.
- You need to know the wizard password for the Palace in question. If you’re the local god (server owner or Palace operator), or if you have been entrusted with the wizard password for someone else’s server, you’re ready to begin exploring the wizard powers.

You become a wizard by entering wizard mode via The Palace client.

▼ To access wizard mode

1. Start your Palace client.
2. Connect to the Palace server where your wizard password is applicable.

3. Select **Wizard Mode** from the Palace client's **Options** Menu (select **Wizard** for Macintosh). A window will appear, asking you to type in the wizard password.
4. When you enter the correct password, a new menu — the **Wizard Menu** — will be added to your menu bar.

Wizard mode remains in effect until:

- You turn it off,
- You disconnect from the current server, or
- A local god turns the wizards permission off.

NOTE – Your wizard status is not saved anywhere on the server. If you want to use wizard or god powers next time you sign on, you will have to enter the password again. This is a security feature; it keeps others who use your Palace client from automatically becoming wizards themselves.

Becoming a god

In a sense, gods are “superwizards” – they possess all the powers of wizards, plus more. Like a wizard, you enter god mode through your Palace client.

▼ To access god mode

1. Through your Palace client, connect to the Palace server where your god password is applicable.
2. Select **Wizard Mode** from the **Options** Menu (select **Wizard** for Macintosh). A window will appear, prompting you for the wizard password.
3. To become a god on the current server, type the god password into this window instead of the wizard password.
4. Whichever password you enter, the **Wizard Menu** will appear on your menu bar.

Your godhood will remain in effect until:

- You exit wizard mode, or
- You disconnect from the current server.

Using the Wizard menu

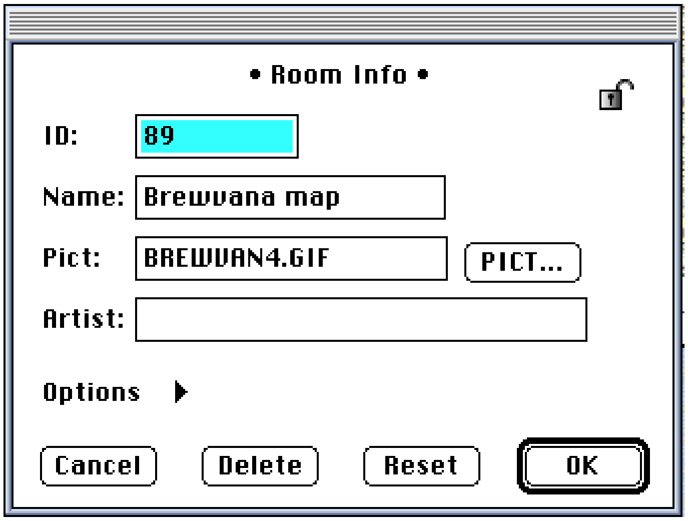
The Wizard menu has the following options.

New Room

This menu command causes the server to create a new room and add it to the script on the server (this may take a minute on slower systems). The new room will receive a *RoomID* number one higher than the highest previous *RoomID* used. Its default name will be "New room <*roomID*>," and its default background will be “`Clouds.GIF`.” Your avatar will appear in the room.

Room Info...

This menu command calls up the **Room Info** window. This window displays a number of important room parameters that you can change while in authoring mode. These serve as “shortcuts” for key Iptscrae commands, which cause the server to alter the script for the current room.



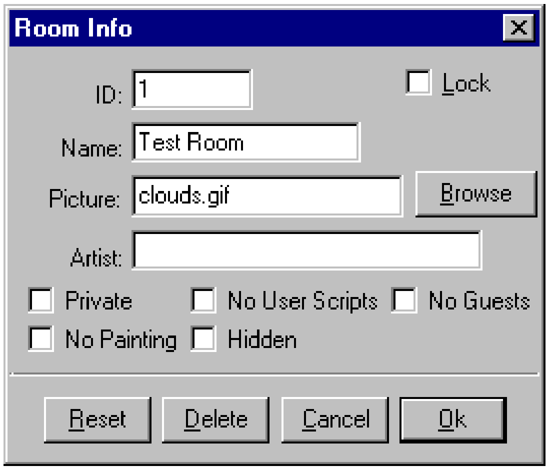
• Room Info •

ID: 89

Name: Brewvana map

Pict: BREWVAN4.GIF

Artist:



ID: 1 □ Lock

Name: Test Room

Picture: clouds.gif

Artist:

□ Private □ No User Scripts □ No Guests

□ No Painting □ Hidden

The Room Info window: Macintosh (top) and Windows (bottom) versions

The **Room Info** window has the following buttons and icons:

- To reset, delete, or cancel, click the appropriate buttons at the bottom of the screen.
- To accept your values, click **OK**.
- To lock your settings, click either the lock icon (for Macintosh) or select the Lock field (Windows). This locks the settings of the **Room Info** window, denying access to the room script, by creating a special password for the room information. Once locked, no wizard or god will be able to change these settings without first entering this special password (not even you).

The **Room Info** window has the following fields. Iptscrae equivalents are noted.

- **ID**

  The identification number assigned to the current room appears here. You can change the ID by typing into this box. Iptscrae Equivalent: `ROOMID` *`number`*

NOTE – This password, like all Palace-related passwords, is saved in an encrypted format in the server's script file. The fact that these passwords are encrypted protects their integrity even if your script is shared or otherwise distributed. If you forget one of your passwords, you can remove the lock by deleting the line responsible for locking the room in the first place. To make this change, simply open the script file with any text editor and delete the appropriate line. It will look like this:

```
LOCKED “weird_garbled_password”
```

- **Name**

  Enter a name for the room. Iptscrae Equivalent: `NAME "`*`roomName`*`"`
- **Picture**

Enter the name of a background picture for the room. If you are unsure of the exact name or location of the graphic you want to use, click **Browse** (Windows) or **PICTs…** (Macintosh). Background pictures in The Palace must be 512 by 384 pixels in size, and must be saved in GIF format (but see the following note). In addition, these images should all use the official Palace Palette, which can be extracted from any of the original Palace artwork,or downloaded from various Palace-related websites. This is because your images will be displayed in the Palace Palette even if they were created using different palettes, and will probably undergo some color shifting as a result. The best way to avoid this is to use the Palace Palette from the beginning, whenever you create original artwork for your site. Iptscrae Equivalent: `PICT "picture.gif"`

NOTE – The Palace client for Macintosh (version 1.1 and above) can handle .JPG files as well as .GIFs. To support older Macs and Windows users, however, GIF images must be still made available. When entering a room for which both file types exist, these advanced Macintosh clients will request the JPG file only, while older Macs and Windows clients will request the GIF file only.

- **Browse** (Windows version)
- **Pict…** (Macintosh version)

  This button calls up a window allowing you to search your system for graphics to use in the Picture box. The name of the file selected here will be loaded into the Picture box automatically.

- **Artist**

  Enter the name of the artist(s) responsible for the art used in the room. Currently the contents of this control are accessible only to wizards. Iptscrae Equivalent: `ARTIST "`*`artistName`*`"`

- **Private**

  Check this field to hide the number of people in the room. Generally, when a user looks in the **Rooms** window (via the client's **Options** Menu), each room is shown with a number representing the number of people currently in that room. Private rooms, on the other hand, possess a dash instead of a number. The only way to tell the number of users in a private room is to enter it. Iptscrae Equivalent: `PRIVATE`

- **No Painting**

  Check this field to prohibit painting/drawing in the room. It is a good idea to set painting off in rooms with a large amount of scripted activity (looping alarms, for example); this is because painting can seriously affect the speed of events in the room. Iptscrae Equivalent: `NOPAINTING`

- **No User Scripts**

  Check this field to prohibit user scripts (also known as ***Cyborg Scripts***) in the room. It's a good idea to turn user scripts off in rooms that already possess a lot of scripted activity (continual `SETALARM-ON ALARM` loops, for example). You might also want turn scripts off in rooms where large numbers of people are expected to gather, as these scripts can seriously affect the speed of events in the room. Iptscrae Equivalent: `NOCYBORGS`

- **Hidden** (Windows)
- **Hidden From Room List** (Macintosh)

  Check this field to prevent the room from showing up in the **Rooms** window. For all intents and purposes, the room doesn't exist for non-wizards (wizards and gods still see all rooms in the Palace). Iptscrae Equivalent: `HIDDEN`

- **No Guests** (Windows)
- **No Guests Allowed** (Macintosh)

  Check this field to prohibit guests from entering the room (making the room into a Members-Only room). Iptscrae Equivalent: `NOGUESTS`

New Door

Select this menu option to create a new door in the room. The new door appears as a large square in the middle of the screen, and must be moved to the desired place and size. The new door automatically receives a unique ID number, one higher than the highest ID already used on a door in the current room. Your new door is considered by default to be a ***Passage*** to another room (hence the name of this menu command).

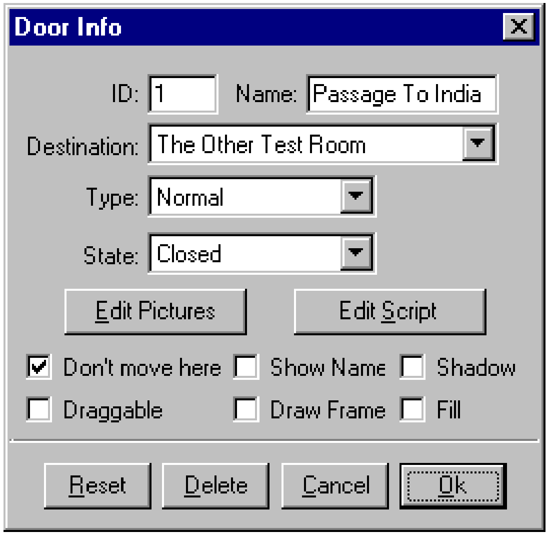
NOTE – ***Doors*** belong to the general object class known as ***Spots*** (short for "hotspots"). In other words, a door is a special type of Spot. As a result, the two words are often used loosely or interchangeably. The differences, however, are pretty significant. True doors are also known as ***Passages***. They specialize in allowing movement between rooms, and possess built-in "door-like" features (i.e., they can be opened and closed, locked and unlocked, and passed through). In addition, they cause the cursor to change to a "pointing hand" when they're rolled over. However (and most importantly) they cannot run scripts.

Spots (of any non-Door variety) can run scripts, but possess no built-in responses for opening/closing, locking, and other door-like behavior. While much of this behavior can be mimicked with Iptscrae, there is currently no way to make a Spot change the mouse cursor in response to a "rollover" event.

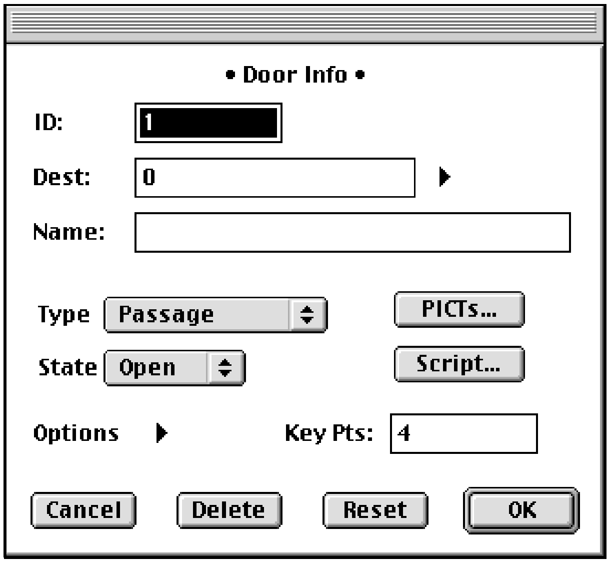
Door Info...

This menu command calls up the `Door Info` window for the currently-selected door/spot, and can also be accessed by double-clicking on the door/spot in question. The window displays a number of important parameters that you can change while in

Authoring Mode (from the `Authoring Mode` menu option). Like the **Room Info** fields, these fields are actually shortcuts for Iptscrae commands, causing the server to alter the script for the selected door/spot.



ID: 1
Name: Passage To India
Destination: The Other Test Room
Type: Normal
State: Closed
☑ Don't move here ☐ Show Name ☐ Shadow
☐ Draggable ☐ Draw Frame ☐ Fill



ID: 1
Dest: 0
Name:
Type Passage
State Open
Key Pts: 4

The Door Info window: Windows (top) and Macintosh (bottom) versions

The exact names and positions of these fields and buttons differ somewhat from the Windows to the Macintosh version. This window has the following fields:

- **ID**

  The identification number assigned to the current door/spot appears here. You can change the ID by typing into this box.

NOTE – All doors and spots must possess ID numbers, and all ID numbers must be unique within their own rooms. Using the same ID number for two spots in the same room will cause your scripts to behave erratically. Iptscrae Equivalent: `ID` *`number`*

- **Name**

  This textbox allows you to type in (or change) the name of the currently-selected door/spot. Unlike ID numbers, names are optional. Iptscrae Equivalent: `NAME "doorName"`

- **Dest** (Macintosh version)

  If the current selection is a Passage, you will probably want it to have a destination. You can select the destination of a Passage by entering the ID number of one of your existing rooms here. If your current selection is a Bolt, on the other hand, you'll want to set its "destination" to equal the ID number of the Lockable Door it toggles. Iptscrae Equivalent: `DEST` *`number`*

- **Destination** (Windows version)

  Unlike the **Dest** command for Macintosh users, the Windows version allows you to select the door’s destination by name rather than by number. You select the destination of a door by choosing one of your existing rooms from the list that appears when you click on this control. Iptscrae Equivalent: `DEST` *`number`*

- **Edit Pictures** (Windows version)

- **Pict...** (Macintosh version)

  This button calls up a window known as the Pictures List, which allows you to assign graphics to the door/spot in question. These graphics are associated sequentially with the various possible States of the door/spot (as controlled by the `SETSPOTSTATE` and `SETSPOTSTATELOCAL` commands). For more information, see *Authoring* on page 57.

- **Type**

  This control allows you to change your door to a spot, or to another type of door. Clicking here causes a drop-down list to appear, from which you may choose one of the following types:

Type	Description
Normal	a.k.a. “Spot.” A programmable hotspot that will run Scripts.
Passage	Allows automatic passage to the room specified as the Destination (in the door Info window).
Shutable	May be opened or closed, but may not be locked; allows passage to Destination when open.
Lockable	May be locked and unlocked; may be opened if unlocked; allows passage to Destination when open.

Type	Description
Dead Bolt	Toggles the state of a Lockable door; not actually a door itself.
Nav Area	Constrains user movement. Once a user enters, they cannot move out unless they move into another door or spot. In other words, they cannot move onto any “non-spot” areas of the screen. Use sparingly.

Door/Spot Types

- **State**

  Doors and Spots can possess a number of different states. These states are referred to by number: for all Shutable or Lockable doors, a state of 0 (zero) equals **Open**, and a state of 1 (one) equals **Closed**. A door’s state may be changed by using the Iptscrae commands `SETSPOTSTATE` or `SETSPOTSTATELOCAL`. For example, if ID number 3 corresponds to a Shutable Door, you can shut it by entering the following command via the input box in The Palace client:

  ```
  / 1 3 SETSPOTSTATE
  ```

  For most doors, **Open** and **Closed** cover the territory pretty well, but “Normal doors” (a.k.a. "Spots") may possess up to 32 states, beginning with state 0 and ending with state 31. If, in the above example, ID number 3 corresponded a Normal Spot, the same line of input typed above would cause it to enter “state 1,” which might have all kinds of results depending on the script attached to it. Most often, such state-changes are used to change the graphic displayed in the spot.

  Since a spot’s state remains the same until it is deliberately changed (even if you leave the room and come back), it may be used as a kind of room-level “global variable.” You can determine the current state of any spot or door by using the `GETSPOTSTATE` command. The following line of code, for instance, will cause a spot to move one step forward through a three-state loop (0 - 1 - 2 - 0 - 1 - 2 - etc.) each time it executes:

  ```
  { 0 ME SETSPOTSTATE } { ME GETSPOTSTATE 1 + ME SETSPOTSTATE }
  ME GETSPOTSTATE 2 == IFELSE
  ```

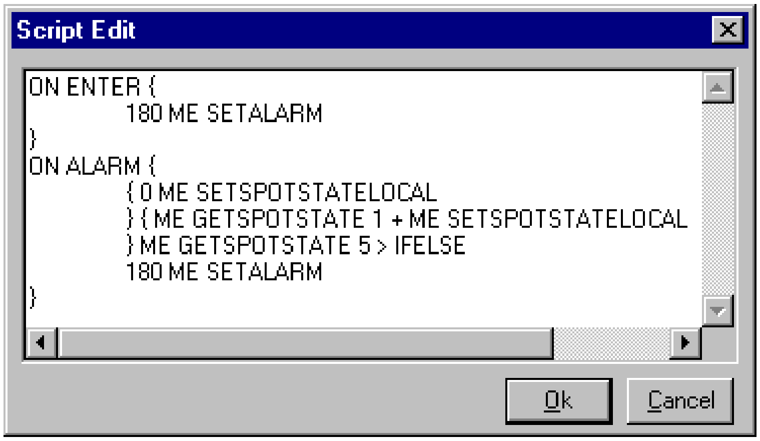
  Within an actual script file, this line would usually be written out in an indented format as shown below, making the code a little easier to read and maintain. It would also be placed within an event handler such as ON ALARM:

  ```
  ON ALARM {
     {
        0 ME SETSPOTSTATE
     } {
        ME GETSPOTSTATE 1 + ME SETSPOTSTATE
     } ME GETSPOTSTATE 2 == IFELSE
     60 ME SETALARM
  }
  ```

- **Edit Script** (Windows version)

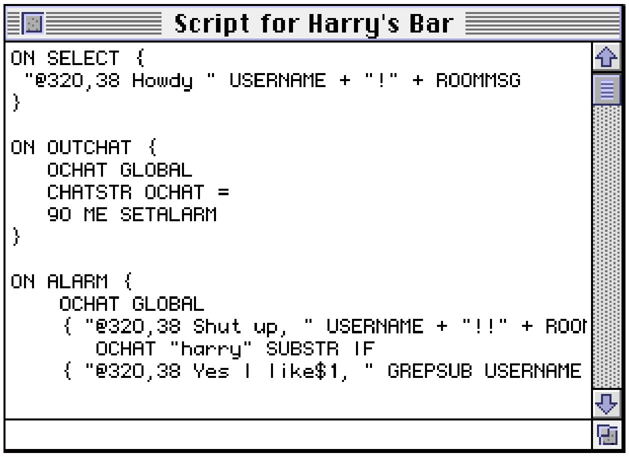
- **Script...** (Macintosh version)

  Click this button to display the scripting window, where you may edit the part of your script that corresponds to the currently-selected spot. Be sure to make a backup copy of your script before editing it in any way.



```
ON ENTER {
        180 ME SETALARM
}
ON ALARM {
        { 0 ME SETSPOTSTATELOCAL
        } { ME GETSPOTSTATE 1 + ME SETSPOTSTATELOCAL
        } ME GETSPOTSTATE 5 > IFELSE
        180 ME SETALARM
}
```

The Scripting window (Windows version)



```
ON SELECT {
 "@320,38 Howdy " USERNAME + "!" + ROOMMSG
}

ON OUTCHAT {
   OCHAT GLOBAL
   CHATSTR OCHAT =
   90 ME SETALARM
}

ON ALARM {
    OCHAT GLOBAL
    { "@320,38 Shut up, " USERNAME + "!!" + ROOM
       OCHAT "harry" SUBSTR IF
    { "@320,38 Yes I like$1, " GREPSUB USERNAME
```

The Scripting Window (Macintosh version)

Door Options. Windows users can access various options directly, by using the checkboxes in the **Door Info** window. For Macintosh users, however, this window has a button labeled **Options**. Clicking on it brings up the **Door Options** window, allowing Macintosh wizards to set options for the selected door. Regardless of how you access them, the effects of these options are the same for both Windows and Macintosh users.

- **Don’t Move Here**

  This control, when checked, keeps users from moving onto the door/spot.

- **Draggable**

  Currently unimplemented.

- **Show Name**

  This control, when checked, causes the name of the door/spot (as explained above) to appear superimposed on the image in the View Screen. This is especially useful for creating links to other Palaces or web pages.

- **Draw Frame**

  This control, when checked, causes the outline of the door/spot to be visible to all users. It appears as a thin black line.

- **Shadow**

  Currently unimplemented.

- **Fill**

  Currently unimplemented.

Layering

This command brings up the **Layering** Submenu, which contains four choices: **Send To Bottom, Send Back**, **Bring Forward,** and **Bring to Top**. These cause the program to literally change the placement of the selected door/spot in the script, otherwise known as the “Z-Order.” The Z-Order is the order in which spots are laid down; it is derived from the order in which they appear in the script (which does not necessarily follow their ID numbers). There are two main reasons for using these commands. First, they allow you to select a spot even if it’s buried beneath another one. Second, when one spot does overlap another one, any graphics associated with it will cover up the other spot’s graphics. By pushing the top spot beneath the bottom one (“Send to Back”), you cause the associated graphics to swap places as well.

Changes made via the Layering commands are retained in the server script. To test this function, try the following:

1. Make a room with two overlapping spots called *Spot1* and *Spot2*. Spot2 should be on top.
2. Select **Authoring Mode** from the **Wizard** menu to go into authoring mode, and associate a graphic with each spot. The graphic for Spot2 will overlap the graphic for Spot1.
3. Select Spot2 and select **Send to Bottom** from the **Layering** Submenu. Immediately, the graphic associated with Spot2 will be overlapped by the graphic for Spot1.
4. Exit the Palace, kill the server and examine the script file with any text editor. You’ll see that Spot2 now appears before Spot1 in the script.

Authoring Mode

This menu item causes you to enter authoring mode. In this mode, the outlines of all doors and spots become visible to you, and clicking on them does not activate them. Instead, until you turn authoring mode off, clicking on a door or spot merely selects it for subsequent operations; you can double-click on any door or spot to bring up the **Door Info** window for that door. For Windows users, the hotkey combination for this command is Control-A. For Macintosh users, it’s Command-` (that’s a single backquote).

3 Wizard and God Commands

The following section describes the syntax and function of all special commands available in wizard mode:

- *Wizard commands* on page 29
- *The god commands* on page 34
- *Message shortcuts* on page 39
- *The `page command revisited* on page 40

Wizard commands

While in wizard mode, you have access to a number of wizard commands. These commands are entered just like any other special commands (i.e., they're typed directly into the Input Box, as though you were speaking). All wizard commands use a forward quote (′) or a backquote (`) as their initial character (either may be used).

NOTE – ***Targeted commands.*** Many wizard commands may be “targeted,” meaning that they will take effect only upon a particular user. You can activate these commands on a user by clicking directly on the person’s avatar. That user’s name or user ID will be passed as an argument to any target command subsequently typed into the Input Box. For example, to gag someone named "Darth," you could simply select Darth (as if selecting them for private chat), then type `` `gag `` and press the ENTER key. All targeted commands are noted.

The following listings describe the syntax, alternate uses, and effects of all wizard commands.

`autoannounce *message*

This command changes the “autoannounce” message (seen by all users upon logging in).

`banip [*minutes*] *ip*

This command permanently bans all "guest" users coming from the IP address specified *(ip)*. Until the ban is removed by an `` `unban `` command, no one from the IP address specified will be unable to connect to your Palace. This command only accepts numeric IP addresses (not domain names), although wildcards may be used (e.g.: *102.57.34.**). The default duration of the ban is set with the `` `death `` command.

`banlist

`banlist -a *address*

`banlist -t

This command displays the ***Banlist***, which is a list of all users who have been tracked, killed or banned from your Palace, as well as the names and comments of the wizards or gods who banned them. The `-a` option is used to display a list of banned users coming from the specified address. The `-t` option is used to display a list of *tracked* users (see the command *`track [minutes] name* on page 33). The Banlist is displayed in the *Log window*.

`banuser [*minutes*] *name*

This command bans user *name* and keeps them from reconnecting to your site for a period of time equal to *minutes*. If the duration is not specified, the default duration is used. The default duration is set with the `` `death `` command.

`bots on

`bots off

This command allows or prohibits use of “Bots” and `CYBORG.IPT` scripts on the server. Default = on.

`comment *ip text*

This command allows you to insert a comment into a ban or track record.

`death *number*

This command sets the Death Penalty on the server to *number* minutes.

`defaultpaneurl [*panenumber*[*urlstring*]]

This command works with the advertising feature to set the default advertising graphic for the Palace. Use the advertising engine to obtain the urlstring used in this command. See the command *`paneurl <room id> [panenumber[urlstring]]* on page 32 to set advertising graphics for each individual room.

`delete [*roomID*]

This command deletes the specified room from the script file on the server. If no Room ID is specified, the current room will be deleted.

`duplicate

This command creates a duplicate of the current room, and places you in it.

`er

“Emergency Response.” This command instantly transports you to the room from where the last non-wizard `` `page `` or `~page` command was issued.

`extend *name minutes*
`extend *ip minutes*

This command (in either of its forms) extends the existing death penalty on the ban record associated with a user (*name*) or IP address (*ip*).

`flood *number*

This command sets the number of Flood Events that will be tolerated (per second) by the server. Default = 100.

`gag *name*

This command “gags” an obnoxious user, causing the server to ignore everything they say (type). A gagged user also may not draw, move props, introduce new props, or delete loose props. The target will remain gagged — even if they disconnect and connect again — until ungagged by a wizard or god.

You can target this command.

`glist
`glist -k
`glist -n

This command lists the status and IP addresses of all users on the server; otherwise the same as `` `list `` . The `-k` option is used to display the unique numeric *Keys* assigned to each user by the server. The `-n` option is used to display the numeric IP address of each user.

`kill [*minutes*] *name*

This command is used to "kill" (disconnect) a user. Despite what it sounds like, a `` `kill `` is not as bad as a `` `ban ``. The name argument is optional; you can do the same thing by specifying the target with a mouse-click (as if selecting them for private chat), and then typing `` `kill ``. The minutes argument is also optional, allowing you to specify the duration of the kill period in minutes (the default duration is set by the `` `death `` command). Of course, a user who has been "killed" can always reconnect eventually, which leads to the `` `ban `` command (see *The god commands* on page 34).

You can target this command.

`list
`list -d
`list -o
`list -p
`list -w
`list -k
`list -n
`list ip
`list *name*

This command creates a list of the Status, Usernames and IP addresses of all users in the current room, or all users specified by argument, which appears in the *Log window*. The `-k` option is used to display the unique numeric *Keys* assigned to each user by the server. The `-n` option is used to display the numeric IP address of each user.

The ip version of the command displays all users matching the IP address (or block) specified. The *name* version of the command displays the status, username, IP address and current room of user (*name*).

`paint on
`paint off

This command allows or prohibits using the Painting Tools anywhere on the server.

`paneurl *room id* [panenumber[*urlstring*]]

This command works with the advertising feature. It sets the url for a pane in room to urlstring. Your advertising engine sets the url for each advertising graphic.

`pin *name*

This command “pins” an obnoxious user, causing their avatar to be sent to corner of the room, wearing a graphically-depicted chain and padlock. The pinned user will be unable to move or leave the room (except by disconnecting) until he or she is unpinned by a wizard or god (see the `` `unpin `` command). You can target this command.

`propgag *name*

This command puts a propgag on user (name), removing any props he/she is wearing. The subject will remain in this state — even if they disconnect and connect again — until unpropgagged by a wizard or god (see *`unpropgag*).

You can target this command.

`purgebanlist

This command purges expired ban records from the server.

`purgeprops *number*

This command purges all props on the server older than (*number*) days. The number specified must be greater than 0. The default value is set by the `` `purgelimit `` command (see *The god commands* on page 34).

`repage *message*

This command sends your message to the last non-wizard who issued a `` `page `` command.

`roommaxguests *number*

This command allows you to set (or change) the maximum number of guests who will be allowed in the current room at one time.

`roommaxocc *number*

This command allows you to set (or change) the maximum number of visitors who will be allowed in the current room at one time.

`sortprops

This command sorts the prop file index so that the most recently-used props are near the top of the file. Assuming that the most recently-used props are also the most currently-popular ones, this will reduce the amount of lag associated with prop searching on the server.

`track [*minutes*] *name*

This command causes a message to be sent to all wizards whenever user (*name*) signs on. The tracking period lasts for the duration specified by the `` `death `` command, unless a timeout period in minutes is provided with the minutes argument.

NOTE – This command works only if the tracked user is online at the time it is issued. Tracked users are listed when you do a `` `banlist `` command.

You can target this command.

`trackip [*minutes*] *ip*

This command tracks all users logging in from the specified IP address until the track is removed by an `` `untrack `` command. This command only accepts numeric IP addresses (not domain names), although wildcards may be used (e.g.: `102.57.34.*`). The tracking period lasts for the same amount of time as specified by the `` `death `` command, unless a timeout period in minutes is provided with the *`minutes`* argument.

`unban *name* | *ip*

This command removes a `` `ban `` on user (*name*) or IP address (*ip*). You can target this command.

`ungag *name*

This command removes a `` `gag `` on user (*name*). You can target this command.

`unpin *name*

This command removes a `` `pin `` on user (*name*). You can target this command.

`unpropgag *name*

This command removes a `` `propgag `` on user (*name*). You can target this command.

`untrack *name*

This command closes the track record on user (*name*), and stops tracking that person. You can target this command.

`wizcount

This command tells how many wizards and gods are currently logged on. You can target this command.

The god commands

The ultimate power in a Palace site rests in the hands of its god(s), who are generally the same people as the owners/operators. Godhood brings with it a share of responsibility as well, since it is the god of the site who ultimately must take responsibility for the look, feel, mood, pace, and ongoing activities associated with the site in the minds of others. To shoulder this cosmic responsibility, Palace gods possess all of the *wizard commands* explained above, plus a special set of *god commands*. These commands provide Palace gods with the means to control access, distribute powers and change server preferences remotely.

God commands are input just like any other special commands (i.e., typed directly into the *Input Box*, as though you were speaking). All god commands use a forward quote (`'`) or a backquote (`` ` ``) as their initial character (either character may be used).

The following section explains the syntax and function of all god commands. Use these commands wisely; the Internet votes with its feet!

Like wizard commands, certain god commands can be “targeted,” meaning that they will take effect only upon a particular user. These “targeted commands” — as indicated by the targeted roundhead graphic shown here — may be “aimed” by clicking directly on the person’s avatar. That user’s name or user ID will be passed as an argument to any target command subsequently typed into the Input Box. For example, to ban someone named "Darth," you could select Darth (as if for private chat), then type `` `ban `` and press the ENTER key.

`author on
`author off
`authoring on
`authoring off

This command (in either of its forms) determines whether or not wizards can enter ***authoring mode,*** allowing them to change room settings, create new rooms and doors, etc. Default = on.

`avatarurl http://*server*/*directory*/

The `avatarurl` command points to the location of your server defined avatars. The server defined avatar feature is available in Version 4.0 and later of the Palace server. With the use of this command, visitors to your Palace server will be able to use your server defined avatars.

`ban *name*
`ban *userID*
`ban *activation code*

This command (in either of its forms) is used to permanently banish a particularly obnoxious user. The banned person won't be able to connect to your server again until you decide they've been rehabilitated. The maximum possible duration for a `` `ban `` is 32,767 minutes. If you use the name parameter, that user will not be able to log on under another name. They must, however be connected to the server to be banned by name.

`botkill on
`botkill off

This command determines whether or not the `KILLUSER` command can be used by Bots (Cyborg Scripts) on the server. Default = off.

`chatlog on|off [*filename*]

This command toggles the server's *Chatlog* on and off. The Chatlog file contains all chat generated by all users on the server. The filename argument is optional.

`custom on|off

This command determines whether or not members may use custom props on the server. The default is on. Set this command off if you want visitors to your site to only use the server defined avatars.

`defaultroomocc *number*

This command allows you to set (or change) the default number of users who will be allowed in any given room at one time. This is a server-wide setting; it does not change the occupancy of rooms in which the maximum occupancy has been set individually.

`downlist [*number*]

This command moves the current room down in the room list (i.e., toward the end of the script file) by the number of rooms specified.

`dropzone on|off

This command enables or disables the current room's ability to serve as a dropzone. A dropzone is an alternate entrance; a room where incoming users may appear if the actual Entrance room is too full to admit them. In such a case, the server looks through the script file for alternate entrances, placing the user in the first dropzone it finds (with vacancy).

`fileserver *server/directory/*

This command allows you to serve your room pictures from a web server, rather than from your Palace server.

`flushpropscc

This command flushes props older than n days from the prop file. This should be part of your periodic Palace server maintenance.

`godpass *password*

This command allows you to change the god password.

`guests on|off

This command determines whether or not guests (unregistered users) will be allowed to sign on to the server. Default = on.

`maxocc *number*
`maxserverocc *number*

This command (in either of its forms) allows you to set the maximum number of users who will be allowed on the server at one time. Note that the maximum value of `number` is automatically limited by your license capacity; this command may be used to set it anywhere up and including to this licensed maximum.

`memberrooms on|off

This command, when on, allows members to create their own rooms. Default = off.

`nowhisper on|off

This command toggles all users' ability to whisper (i.e., private chat) on the server.

`passwordsecurity on|off

This command toggles the state of the server's password protection function. If on, the server will automatically kill any user who enters an incorrect password (wizard or god) three times in a row.

`picdir *relativePath*

This command allows you to set the location of the *pictures directory* (folder) on the server, which tells incoming Palace clients where to find any images they don't already have. If the `picdir` is set to an incorrect path, new users will not receive any downloaded files, even though users who have visited the site before will see everything perfectly. The pictures directory stores not only all of your graphic images, but all other files intended for automatic download (such as your MIDI or WAV files). For operators of Windows- or Mac-based Palace sites, the effect of this command is the same as opening the Server **Preferences** window and setting the Server Directory. On UNIX-based sites, it is the only means of changing this setting without taking the script offline for editing.

NOTE – ***SYNTAX is Platform-Specific.*** The `` `picdir `` command sets the contents of the `PICFOLDER` line in the server script. The format of this line (and therefore the format of the `relativePath` variable) differs depending on the operating system used by the Palace server. Windows and UNIX servers expect to see forward-slashes *(/)* used as delimiters, as shown here.

Wizard Command:

```
`picdir pictures/
```

Server Script:

```
PICFOLDER "pictures/"
```

Macintosh servers, on the other hand, use colons *(:)* as delimiters. In addition, a colon delimiter must be placed at the beginning of the path, as shown here:

Wizard Command:

```
`picdir :pictures:
```

Server Script:

```
PICFOLDER ":pictures:"
```

A final delimiter is <u>always</u> required, regardless of server platform.

NOTE – ***PATH MUST BE RELATIVE.*** It is important to note that the picfolder must be specified in terms of a <u>relative</u> path; that is, starting from the location of the main "Palace" directory. For instance, if your Palace server resides in a directory called `/local/thepalace` and your pictures are located in `/local/thepalace/media/pictures`, the correct wizard command to set the picfolder would be:

Windows/UNIX Servers:

```
`picdir media/pictures/
```

Macintosh Servers:

```
`picdir :media:pictures:
```

`playerkill on|off

This command determines whether or not members can use the `` `kill `` command on the server. When `` `playerkill `` is *on*, any member (or any member's Cyborg Script) can boot any other user off the system. Default = off.

`recycle [*number*]

This command causes Guest Numbers (the individual numbers appended to the word “guest” in Guests’ names) to begin recycling immediately. The number argument causes the guests numbers to recycle after that number. For example,

```
'recycle 1000
```

would recycle the guest numbers at 1000.

`revision

This command gives the revision number and version of the Palace server.

`savesessionkeys

This command toggles whether puids and activation codes are stored in the log.

`securelogoff on|off

This command, when on, tracks logoffs from a Palace server with the authentication option.

`securesite on|off

When this command is set on, a user name and password are required to access the Palace server.

`security host:port

This command tells the Palace server where the authentication daemon is running.

`servername *serverName*

This command allows you to set (or change) the name of your Palace.

`shutdown

This command shuts down the server (as soon as possible, which may take several seconds).

`spoof on|off

This command allows (or disallows) members to spoof (make their words appear from somewhere other than their mouths) by using the `@x,y` option when talking. The default is on.

`uplist [*number*]

This command moves the current room up in the room list (i.e., toward the beginning of the script file) by the number of rooms specified.

`wizards on|off

This command determines whether or not wizards are permitted on the server (gods remain unaffected). Default = on.

`wizardkill on|off

This command determines whether or not the `` `kill `` command can be used by wizards on the server. Default = on.

`wizardsonly on|off

This command sets the current room so that only wizards and gods may enter.

`wizpass *password*

This command allows you to change the wizard password.

Message shortcuts

While in wizard mode you also have access to three special message shortcuts, allowing you to communicate with many users at a time. Like the other special commands, message shortcuts are typed directly into the Input Box in The Palace client. The difference is that message shortcuts use a tilde (~) as their initial character, as opposed to the tickmarks used by most other commands.

The message shortcuts available to all wizards are:

~gmsg *message*

This shortcut sends your message (whatever you type after "`~gmsg`") to *everybody* on the server; regardless of their user class. The exclamation point may be used to create a “shouting” message.

~rmsg *message*

This shortcut sends your message to everybody in the room, regardless of their user class. The exclamation point may be used to create a “shouting” message.

~smsg *message*

This shortcut sends your message to all other wizards and gods on the server. The exclamation point may be used to create a “shouting” message.

The `page command revisited

As mentioned above, the ~smsg shortcut can be used to send a message to all other wizards on the server. But since all Palace members can do the same thing by using a `page command (and the message sent is still visible only by wizards), this second method turns out to be more popular. In fact, many Palace sites use the `page command as a kind of "wizards' CB channel," and it works well for that. For example:

```
Page from Wiz1: Keep an eye on Guest 1641 from 206.17.52.4. I had to
kill him earlier today.
```

Replying to pages

There are two ways to reply to a page. The most common method is the 'er (emergency response) command. This command notifies other wizards on the site that you are responding to the page, and instantly transports you to the room where it came from.

This only works when the page emanated from a non-wizard - e.g. it keeps track of the last *non-wizard* page. For example:

```
** Page from Joe in Red Room: That nasty hacker dude is back.

'er
```

But there are times when your direct presence isn't really necessary, such as when a guest pages you to ask "Where's the help function?" To simply send a message back to the person who paged you, use the `repage ("reply to page") command, like so:

```
** Page from Joe in Red room: wizard Help!!

'repage What can I do for you?
```

Again, the other wizards on the site will be notified that you have responded to the page, as shown below:

```
** Page from System: Wiz1 has responded to Joe's page via 'repage
```

4 The Good Wizard

This section contains a number of guidelines and discussions intended to help you "grow" your site and maintain a positive and effective reputation as a wizard. Some of these guidelines suggest general attitudes and actions to be performed on a continual basis, while others are intended to help you recognize and deal with disruptive or abusive users, in specific circumstances.

Overview

These guidelines were developed based primarily upon the experiences of wizards at the "main" Palace, and may apply differently to your site. In fact it's fairly certain that you will eventually encounter a situation in which these recommendations are ineffective or inappropriate. When this happens, you'll have to rely upon your own creative solutions, and let your own experience be your guide. Since all wizards are individuals and have their own styles, this is probably for the best.

While reading this section, it is important to remember that these are *guidelines* and *suggestions*, not *laws* or *rules* of any kind. Not only is each one entirely optional, but even the ones you decide to follow will probably be "stretched" from time to time. In other words; they're not written in stone, and they're up to you to enforce or ignore as you see fit.

Roles and responsibilities

There are many common misconceptions about wizardship. One of the most common is the idea that wizardship is a "reward" given to people who have been using The Palace software for a long time, or hanging out in a particular Palace since its opening. While familiarity with the software and knowledge of the site's culture are indeed important for every wizard to possess, these traits can be picked up very quickly given a short training period. What can't be learned easily are the "softer" aspects of the wizard's role; the *social graces,* if you will. Empathy, diplomacy, charisma, wisdom, and discretion — these are the kind of skills that are most important for successful performance of the wizard's duties.

Another common misconception is that wizardship will be granted to you if you ask for it often enough. Nothing could be further from the truth. In fact there's a fairly reliable rule of thumb among Palace operators: anyone who asks to be made a wizard probably shouldn't become one. Indeed, the more a person requests wizardship the more they prove themselves incapable of handling social situations rationally and creatively, which is exactly what wizards must excel at. Requests for wizard status occur more often than you might expect (they're practically a daily routine on large public sites), and from the standpoint of the Palace operator they resemble nothing so much as childish whining: *"Look at ME!" "That's not FAIR!"* and *"Are we THERE yet?"*

In the sage words of Palace inventor Jim Bumgardner: "Wizards exist for one principle reason: to help make the Palace a better place."

Obviously "better" is a relative term, and each Palace site will define it differently, so we should append the phrase "for the majority of your site's users." This goal is accomplished by employing a variety of proactive and responsive techniques, the most important of which are listed below (in rough order of frequency):

- Being your usual brilliant, entertaining self.
- Talking to users.
- Reprimanding abusers.
- Gagging obnoxious abusers.
- Killing extremely obnoxious abusers.
- Occasionally making minor modifications to rooms (depending on the permissions and rules set by the local Palace gods; many Palace sites discourage use of *authoring mode* on the server).

There's one more thing all wizards must possess; one trait that has no relation to the other skills mentioned above. That thing is *time*. Becoming a wizard is a *commitment*, not a membership card to an exclusive club. When you are granted wizard status at a Palace site you become part of an "elite team," dedicated to making sure the site runs smoothly and the party doesn't get out of hand. Depending on the popularity of the site and the hours set by your Palace operators, it can get to be a full-time job. Before you don that wizard badge, make sure you have the time to fit it into your life.

Setting an example

It may not seem apparent at the time, but every time you do something in The Palace you are defining the social order there: you are contributing to the overall conception of "the kind of things that happen at your site." This is one reason why it's a good idea to have a wizard online at all times to provide quick responses to ~page commands, and why security actions must be swift and precise. If you're like most wizards (and gods), you don't want people to have give the impression that your Palace is a chaotic and lawless place, but you're not trying to create a police state, either.

Due to your rank and power, your effect on the local culture is considerable; it should not be underestimated or wasted. This cuts both ways: obviously, if you set a good example and act responsibly, "good people" will reflect your attitude and "bad people" will be quickly dealt with. New users, finding your Palace a friendly place to learn about the software, will hang around longer and your "population" will gradually increase. But what may not be so obvious is the other side of the coin. If you act in a negative or confrontational manner, killing people without warning or using questionable language, you shouldn't be surprised to find that your users quickly begin emulating your behavior, or that those few who enjoy that sort of thing are the only visitors you have left. Soon gangs, fanatics, and guys named "psykoK!LL3R" will be popping up like weeds, using your site as their own personal battlefield. This kind of environment can send your site into a tailspin as attitude clashes and personal grudges develop into 24-hour-a-day flame wars.

Helping new users

As mentioned above, one of the things that can add a lot to a Palace's appeal is the willingness on the part of the wizards to help new users learn about The Palace software in a friendly, supportive environment. No one is born knowing all the functions in the interface, and — like it or not — few find the time to read the manuals thoroughly, if at all. Besides, for many people, the best way to learn something is just to jump right in and play with it, as long as they can ask questions of a more experienced person as they go. The Palace interface is simple enough that this approach works very well for most people, and as a result a lot of your visitors may be first-time users of the software.

Palace users tend to roam from site to site, depending on Internet traffic, special events and "what's hot this week," but most of them soon find a site or two that they think of as a "home base." Very often, this home base is one of the first sites they ever visited — or the first one where the people were friendly and there was a good conversation going on. By helping new users with their problems and learning curves, you increase your chances of becoming that site for them.

The host with the most

But the world isn't all about security and learning curves. In addition to a place where they feel safe, people want a place to have *fun*, and *that's* something The Palace can provide in spades. A good Palace site is something like a public park (or a private club, if you prefer), in which a great party is always going on. Ideally, you want your users putting your address in the first slot on their bookmark list, running around talking about how much fun your Palace is, and calling up their friends, telling them to join the party!

You certainly wouldn't expect guests to stay long at a party where nothing was happening, and unfortunately, on a Palace site you can't even offer free food to seduce them to stay. Unless your site is extremely popular or comes with its own built-in user group, it's often a good idea to provide some kind of special reason for hanging around. Whatever your special draw is — cool props, animated rooms, interactive scripts, funny sounds or beautiful art — one thing you should always try to maintain is a relaxed and friendly atmosphere, a place where people feel free to be themselves (or their alternative selves), and join in a growing sense of community.

Like any party, the initial "feel" or "theme" is set by the host. Certainly, everything about your site — from art and sounds to scripts and custom props — contributes to the overall feel of the place, but as a wizard, it's helpful to remember that *you* are the *living* part of the site. If you're a fun person to be around you'll attract good company, and people will be stopping by to visit long after the appeal of the interactive doodads has worn off.

Special events

Another way to increase the fun factor (and population) of your site is to hold special events of various kinds. Special events come in all shapes and sizes, and may be run on a regular or occasional basis. With the proper scripting, The Palace software can support a variety of interactive games, puzzles and "exploratory environments," as well as more free-form activities such as cooperative painting, room decorating (a common pastime in some circles) and the ever-popular "avatar contest."

The most important thing about special events is that you let people *know* about them.

Special events on your Palace site should be mentioned in the *Announcement* field of your *Palace Directory* listing. This field resides in the script file itself, and cannot be changed via the client. There are two ways to change the Announcement field. The first is to shut the server down and edit the script file directly (on UNIX systems this is the only way to change the Announcement). Windows and Macintosh users have another option, which doesn't require shutting down the server. For these operators, the Announcement field can be accessed via the **Palace Directory** option on the server's **Options** Menu. For more information on the Palace Directory, see *The Palace server Guide*.

If your events begin drawing a real crowd, eventually they'll grow too large to hold in a single Palace room. At this point, you might want to consider beefing up your software. The Palace Inc. offers a special Palace Plugin called the PalacePresents Moderator. With the help of this plugin (and an active wizard password), you can run large-scale "broadcast" events such as interviews, lectures, web tours, and multimedia presentations of all sorts. Using a companion plugin called the PalacePresents Viewer, participants can watch the event in a separate "stage" window — from any room in your Palace site. They can even submit questions or comments to the stage, to be used or discarded by the event moderator (that's you). The PalacePresents Viewer is distributed with the 32-bit Windows version of The Palace client, and is available for free from The Palace Inc. All of the software and documentation required to run PalacePresents events may be found at The Palace website (`http://www.thepalace.com`).

Palace patrol

Perhaps if we lived in a perfect world, we wouldn’t have any need for *virtual* ones. However, such is not the case, and you can pretty much accept it as a fact that — sooner or later – someone will do something in your Palace that you'd prefer they didn’t.

When it comes to maintaining any type of behavioral standards, censorship, or other form of social control over your own Palace, the choices you make are entirely your own.

This is good news on the surface; after all, it guarantees your freedom to run the place the way you see fit as long as you do no one any harm, just like the owner of any other type of establishment. However, it also guarantees that *you* will be the one responsible for creating whatever social standards *will* exist in your Palace, and for communicating your goals and intentions to your users.

Unfortunately, not everyone is considerate (or even aware) of your intentions. The Internet can be a scary place for site owners in these days of “political correctness” and “communications decency,” and the situation is made no better by allowing abusive individuals to waste your bandwidth and drive away your guests. Fortunately, there’s something you can do about it...

Security functions

One of the unfortunate aspects of the wizard's duties is that once in a while it becomes necessary to exercise force when dealing with an uncooperative Palace abuser. Although as a wizard you certainly possess the tools necessary to "kill" users for the least indiscretion, the fact remains that *violence* — even in cyberspace — has a tendency to *escalate*. Killing should therefore be considered only as a last resort (obviously, this excludes "adventure" sites and games in which dying is part of the rules).

Of course, in such a highly subjective area as "social control," there's no way to describe all of the possible cases and mitigating circumstances that may arise in real-time, and we wouldn't want to tell you how to run your Palace anyway. Instead, the goal of this section is to provide a loose set of guidelines that we at The Palace Inc. have found useful in making such judgment calls, and to familiarize you with the security tools at your disposal.

Make chat, not war

Some users are so clueless they'll do things that can be interpreted as being intentionally disruptive, even though they don't intend them that way. Some new users, for example, habitually hit the return key twice (which is a vestige of IRC and BBS chat). There are many similar "minor offenses" that should hardly warrant a wizard's attention, except for the fact that they *do*. Users like this are in need of an *education*, not a reprimand.

In The Palace, as well as in real life, it is always better to *talk* than to *kill*. Not only does killing increase the chance of a disruptive user becoming a truly hostile one, but each kill adds a little more to your reputation as a "mean" wizard (or at least a "hardcore" one) — and wouldn't you rather be known for the *good* things you do? If the "kill ratio" is very high in your Palace, due to intolerant wizards or deadly censor scripts, the user community will learn about it, and the decreasing flow of traffic to your site will reflect this fact. The Palace is not intended to be a combat zone. In fact, even *getting* killed on The Palace isn't very entertaining — it's rather anticlimactic.

Even an obnoxious user should generally be warned at least once before being killed, and given sufficient opportunity to amend his or her behavior. Having said this, there are some cases where you might have to kill without warning. For example, an obnoxious user might be flooding the server by continually repeating the same message (using the up arrow and pressing *Enter* over and over again). In this case it may be difficult to warn the user, because their keypressing nonsense is using up all the available bandwidth (of course, depending on the server's sensitivity to Flood Events, the system may take care of this for you). Another case in which unannounced killing might be appropriate would be that of a previous offender who returns to your site and shows signs of continuing the same behavior. Like everywhere else in life, context is everything.

Just the facts, ma'am

Perhaps only one rule applies equally well to all situations, and it is this: whenever you're on duty, or at least whenever you respond to a page regarding offensive behavior, you should always make sure you're keeping a log file on your own computer. In the heat of the action, things can be said that become forgotten, confused or distorted after the dust settles. Keeping a log is a good way to back up your performance and decisions with cold, hard facts; providing you with an exact transcript of the entire event.

Tools of the trade

This section describes the "arsenal" of virtual tools available to wizards as part of the security operations at their site.

The Wizard Badge

At certain Palace sites, you may notice some avatars running around with an asterisk (*) preceding their names (e.g., " `*JustBob` "). But if you try changing your name so that it begins with an asterisk, it doesn't work. What's up with that?

The asterisk is a special character: its presence indicates that the person in question is either a wizard or a god, since guests and members aren't allowed to possess names beginning with asterisks. As a result the prepended asterisk has come to be known as a "wizard badge." Note that wizard badges are in no way mandatory; some sites prefer to use them, while others prefer not to. If your site *does* use wizard badges, remember to be consistent with them — wizards on such sites should wear their badges at all times while on duty.

A Wizard by any other name

It is also recommended that wizards consistently use the same name when on duty, unless circumstances dictate that another name would be more effective (e.g., during a "sting operation," or while observing events "incognito"). There are a number of reasons for this. First, most of the time (thankfully), wizards are more like "beat cops" than detectives: their primary duty is to make life easier for everyone on the site, and part of that involves maintaining a visible presence. A lot of people just want to know directions to that room they heard about, or when the next "Wheel of Cheese" game is scheduled. The badge makes it easier for newcomers and other users to find you in the crowded *User window*, and reminds potential offenders that wizards are present.

The `list command

The `list command is used to get fundamental information about users, in terms of their names and IP addresses (both DNS and numeric IP). It may be performed either on an individual or a whole group. This is particularly useful for identifying the return of chronic abusers. In its unmodified format (" `` `list `` "), the names and IP addresses of all users in the current room will be displayed in the wizard's *Log window*. The `` `list `` command can take a number of optional arguments. A few of the more useful combinations are described in detail here.

- **`list Guest 123**

  This variant lists information for Guest 123 only. This is the preferred way to get info on a user, as it makes only one call to the nameserver, instead of one for each user in the room.

- **'list -k abuserName**

  The `-k` variant adds an alphanumeric key to the end of the listing, which is based on the user's registration code. Since this key is linked to the user's copy of The Palace client, it can be used to uniquely identify a particular member regardless of IP address.

- **`list 205.23**

This variant lists info for all users whose IP addresses begin with "205.23." Note that you *must* use a numeric address (as opposed to a DNS name) for this to work. This can be used to help determine whether an apparently new abuser *is* or *isn't* the same one you killed before.

- **`list -g**

  This variant lists all users currently logged on. Its use is not recommended, as it temporarily lags the server.

In all cases, the `list command has the following format:

```
*** ; g ratbot salon.thepalace.com (206.17.52.5) Red Room
*** ; n Guest 679 204.87.136.14 Palace Gate
*** ; m CrashCat slip0.worldnetoh.com (206.21.38.10) Palace Gate
*** ; w Mark J closet.thepalace.com (206.17.52.4) Red Room
*** ; m Lally orlfl2-6.gate.net (199.227.1.134) The Armory
*** ; n Guest 529 la-ip29.via.net (204.188.133.29) The Spa
*** ; m rorschach 206.17.52.6 Palace Gate
```

The single letter in front of the name represents the member's status:

g = god

n = newbie (same as "guest" — the "g" was already taken)

m = member

w = wizard

If the DNS-style address is missing (as in the case of rorschach and Guest 679 above), that means the site's name server couldn't find one. You may be able to find one by doing an nslookup on your local ISP, if you have access to a UNIX shell account, otherwise you might try using an nslookup utility or web-based search engine.

As a general rule, NEVER PUBLICLY EXHIBIT a user's IP address. Palace servers keep this information private as a favor to users. Of course, you *are* encouraged to share this information with your fellow wizards via the ~page command; as long as it's a matter of security, they have a right to know.

Gagging

You can gag a user named "Paul" by typing:

```
'gag Paul
```

If he's in the same room as you, you can *target* him (by clicking on his avatar) and then simply type:

```
'gag
```

Gagging a user prevents him from saying anything to anybody in any room. If you gag a member, the gag will remain in effect *permanently* until the member is either ungagged, or remains logged off for 2 hours. At least one Member has reported a "bug" that turned out to be a gag. If you gag a guest, however, they can simply sign off and sign back on to release themselves, which pretty much begs for the *banip* command to pulled out.

You would ungag Paul by typing:

```
'ungag Paul
```

It's pretty much a matter of personal discretion whether to *kill* or *gag*. Sometimes gagging an abusive user, talking to them and then ungagging them after getting a visual sign of their acquiescence is all it takes to keep the peace. In other cases, however, this simply makes them more obnoxious than ever. In such cases, you have no choice but to issue a kill command. The following guidelines are helpful when choosing whether to gag or kill: if the user isn't using obscene or foul language, if they're simply being an obnoxious pest (repeating the same phrase over and over, etc.), gagging is probably appropriate. If the user is being obscene, the typical response should probably be issuing a *warning*, followed (if necessary) by a *kill*.

It's not as important to warn before issuing a gag command, but your next action should be to explain to the user that they have been gagged, what the effects are, and ask them to indicate visually that they understand. They can do this by donning a specified prop, or by using the number keys to nod up and down. Bear in mind that once ungagged, they may immediately resume the obnoxious behavior. Good luck.

Prop gagging

This command allows you to prevent a member from showing offensive props. Even if their language is non-offensive in and of itself, members who wear obscene or hateful props should be propgagged, like so:

```
'propgag Sexy Sally
```

After propgagging them, you should talk to them about the prop, obtain their agreement not wear the prop in your Palace any more, and unpropgag them:

```
'unpropgag Sexy Sally
```

If the member continues to wear the prop, kill them.

Pinning

The `pin` and `unpin` commands operate similarly to gag and ungag, but have a slightly different effect:

```
`pin Guest 5757
```

This command pins the target to the lower right corner of the screen and puts a "chain" prop over them. The target is not allowed to move, and may not put on any props. Note however that unless the offender is also *gagged*, they will still be able to talk.

```
`unpin Guest 5757
```

This removes the effect.

Pinning is particularly appropriate for people who abuse the "physical space" of The Palace (these offenders are sometimes referred to as "blockers").

Killing

You should always perform three basic steps when killing a Palace user: these three steps are `'list`, `'kill`, and `'comment`. In the following example, we'll assume your name is "Merlin" and you need to kill someone named "hakker".

1. Get a list of their IP address and regkey by typing the following:

   ```
   'list -k
   ```

   The server will reply via the *Log window*. For instance:

   ```
   *** ; m hakker machine.somedomain.com (206.17.52.5)
   Wizard's Study {DFEF}
   ```

   Record the user's IP address (machine.somedomain.com) and regkey (DFEF) in your notebook. The regkey is only provided for members — it gives you a unique alphanumeric key you can use to identify a user, even if they use a different IP address or name.

2. Kill them by typing:

   ```
   'kill 120 hakker
   ```

   The server will respond with:

   ```
   Page from System: hakker [machine.somedomain.com (206.17.52.5)]
    killed by Merlin
   ```

   The number 120 is the death penalty, in minutes. This means that "hakker" won't be able to get back on for 120 minutes (120 is a fairly "normal" death penalty). If the person is being *highly* offensive, you may want to use a larger number; sometimes a penalty of a day or two is in order (for reckoning purposes, there are 1440 minutes in a day). Of course, even after the penalty wears off, it can always be extended using the `'extend` command.

3. Remember to *comment* your kill by typing `` `comment `` followed by your message. For instance:

   ```
   'comment 206.17.52.5 Using foul language, second time killed.
   ```

This is important for purposes of maintaining a clear transcript of events as they happened.

NOTE – It is important to recognize that killing a member typically has more consequences than killing a guest. There are two principal reasons for this. First, when members are killed, the software keeps track of their regkey and prevents them from logging on from ANY IP address. Second, members have paid money to register their copy of The Palace software, and therefore tend to feel that they're entitled to more leeway than guests. While this is in no way true as far as abuse goes, you're more likely to hear complaints from members than guests.

Regardless of their membership status, all users should understand that abuse is abuse, and that you have the right to defend your own site from it. If challenged, you can always point out that all users — members and guests alike — "sign" an agreement not to abuse the system when they first install The Palace on their computer.

Site banning

When you kill a guest, no regkey exists; so the server can only keep track of the user's IP address. If the abusive guest switches computers or gets a new IP address from his or her ISP, they will be able to come back on. However, in such cases you will generally see a pattern to the IP addresses, like so:

```
*** ; n Guest 1223 dal23.why.net (205.240.253.23) Palace Gate

*** ; n Guest 1228 dal37.why.net (205.240.253.37) Palace Gate

*** ; n Guest 1232 dal18.why.net (205.240.253.18) Palace Gate
```

Typically only the last part of the IP address will change. If you believe a guest is a repeat offender, you should do a banip on the entire B block of IP addresses, like so:

```
'banip 120 205.240.253.*
```

In the example above, the "120" is the death penalty, and the "*" indicates a "wildcard" ban of all addresses beginning with 206.17.52. Again, you should comment the ban to indicate who you are keeping out. In this case, you should provide only the first 3 numbers in your 'comment:

```
'comment 205.240.253 Guest 1223 violated at least 3 of the 10
commandments.
```

It is important to note that a "wildcard" ban only works on *guests* from the IP addresses in question; not *members*. Also of note is the fact that this command can be easily overapplied: for instance, although the software will allow you to ban an entire ISP at one time, it's hard to imagine a case where such a broad course of action would be recommended. Keep the wildcards to a minimum, and try to restrict their use to A- and B-level IP blocks (i.e., the rightmost two numbers in the IP address).

The BanList

The server keeps a record of all currently-banned users, user information and comments in the "banlist." To see it, simply type:

```
`banlist
```

Note that this list can get very large, especially on a popular public site. To search the banlist for a specific person or IP address only, simply type:

```
'banlist name
```

or

```
'banlist ip
```

Whichever variant you use, the requested information will appear in your *Log window*. Here's an example:

```
*** ; Flooder  5 min (0) (192.147.170.47) Guest 6389

*** ; Killed   120 min (79) (www.dialup.axxess.net [166.84.192.118])
C.J.

*** ; comment by Merlin: wizard-baiter with an obscene prop

*** ; Flooder  5 min (0) (ppp3.thelab.com [206.42.130.193]) CozyJo

*** ; Cracker  3000 min (2841) (slip.UC.MING.EDU [128.193.5.144]) Guest
30

*** ; Flooder  5 min (0) (198.105.190.109) FerretMaster

*** ; Site Ban 3000 min (2667) (138.47.16.0)

*** ; comment by DoctorIptscrae: Guest was a major pain

*** ; Killed   120 min (0) (dialup12.goofy.com [138.47.16.12]) Guest
6603

*** ; Killed   120 min (0) (206.29.84.142) Bilbo
```

Not counting the wizards' comments, each line in the banlist shows why a given user was banned, for how many minutes, how much time is remaining (in parentheses), and the user's name and IP address at the time of the kill. A few notes on reading the banlist:

- A "Flooder" is a user who was killed by the server for flooding. Flooders are killed for 5 minutes only.
- "Killed" indicates a user who was killed by a wizard using the ~kill command.
- A "Cracker" is a user who was killed by the server for attempting to use an invalid Palace serial number.

- "Site Ban" means an entire range of IP addresses was blocked out using the `banip command (see above). Wildcard site bans are listed with a zero in the last position of the IP address.

NOTE – You should periodically do a `purgebanlist command to rid your banlist of expired records.

The `hide, `mute, and `reject Commands

Even if things never get to the level where wizard commands are called for, there are still plenty of people who take delight in being annoying, intrusive, or disgusting. Sometimes the effect is subjective: two people can simply rub each other the wrong way, or bring out the worst in each other, for no apparent reason. When both are logged on at the same time, everything seems to take a backseat to their petty personal conflict. Sometimes it's even more subtle: certain people seem to continually emit a "creepy" presence, even though no rules of conduct are actually broken. Sometimes it's hard to tell if anything happened *at all:* you may be called to intervene in a situation where one party *claims* to have been harassed by another party, who (of course) vehemently denies that any such thing occurred...

Without coming down in judgment of anyone in such a situation, it's generally helpful to mention *hide* and *mute,* two built-in commands allowing members to protect themselves from other users.

`mute

Despite what users may believe, the mute command operates locally (i.e., on the computer of the member who executes it); its effect is to screen out any cartoon balloons coming from the muted person. To mute a user called "Guest 13," for instance, a member would type:

```
`mute Guest 13
```

To cancel the effect, the user may type:

```
`unmute Guest 13
```

`hide
`hidefrom

This command allows a member to "hide" from appearing in the *Find User window.* There are two varieties: hide and hidefrom. The `hide command hides you from everybody on the server, while the `hidefrom command hides you only from a particular user. Note that neither of these commands works against wizards, who always see *everyone* on the server. The opposite of `hide is `unhide. The opposite of `hidefrom, logically, is `unhidefrom. Examples:

```
`hide

`unhide
```

```
`hidefrom Guest 13
```

```
`unhidefrom Guest 13
```

`rejectesp
`rejectprivate

The `rejectesp command allows members to reject all incoming "ESP" messages from other users. The `rejectprivate command is similar but rejects *all* private messages (this includes ESP messages). When you turn either of these settings on, it remains in effect until you turn it off — even if you log off. Examples:

```
`rejectprivate on
```

```
`rejectprivate off
```

```
`rejectesp on
```

```
`rejectesp off
```

Recommended escalation procedure

The following steps provide a standard operating procedure for handling users who exhibit consistently or increasingly disruptive behavior. This list is sequential in nature: it represents a gradual escalation of your responses to increasingly abusive actions. As such, it is intended to be applied as a series of "steps," in the order shown.

While this escalation procedure may seem lenient to some wizard types (warning the abuser each time a punishment is about to be meted out, for instance), it's usually better to err on the side of discretion than to "shoot first and ask questions later." After all, the only thing worse than having to deal with a Palace abuser is having to justify yourself after the fact. Seen in this light, these admonishments are more for *your* benefit (and the benefit of any onlookers present) than for the abuser's — they are the online equivalent of firing a warning shot into the air and yelling "Freeze!" Since over-reacting can be as bad as under-reacting, the goal is to effectively solve the problem while using no more than the necessary and appropriate amount of force.

Step One: First Warning

The first step is to address the abuser directly, informing them of the house rules and asking/telling them to stop the abusive behavior. Sometimes the abuser is unaware that what they are doing is "illegal." Most of the time they're are quite aware of what they're doing, but they'll stop if they know they're being watched. At this point you may want to specifically mention the local "remedy" for the offense in question. For instance: "If you don't stop swearing you will be gagged for ten minutes."

Step Two: Gag/Propgag/Pin

If the initial warning doesn't succeed in halting the unwanted behavior, the abuser has effectively decided to challenge your authority. Strange as it sounds, sometimes they will even *dare* you to carry out your threats (i.e., "Go ahead, mister wizard — gag me!") It is important not to get personally offended by these taunts: after all, the only thing you really have to lose is your cool, and you want to keep it. It's not a good idea to kick people for purely personal reasons, as this may give you a reputation as a "bully" or "dictator." This is another reason why it's important to have established local guidelines for specific offenses: when you take action in a highly-charged situation, it helps to know that any other wizard on your site would've done the same thing. If the abuser continues the offensive behavior, simply move to the next step as previously announced. Depending on the specifics of the situation, the appropriate response will be a gag, a propgag, or a pin, lasting for several minutes (or until the abuser logs off).

Step Three: Second Warning

Usually a minute or two under a gag will convince an abuser to log off and go bother someone else. It doesn't keep them from coming back, however, and sometimes it makes them worse, as some of them seem to take this as a personal challenge. In either case, if the abuser persists in acting offensively, you should inform them quite clearly that will disconnect them from the server if they don't stop immediately. While we're on the subject, you may not want to use the word "kill"; it can be taken as an idle physical threat. "Disconnect" has the advantage of meaning exactly what it says, but if you prefer a more poetic term, "kick" is always a good one.

Step Four: Kill

Let's face it: any abuser who's willing to escalate things to step three probably isn't going to stop there. Since they've already been warned and have chosen to disregard it, it's time to be true to your word. Remember that killing on The Palace is neither painful nor permanent: it's more like batting a fly away from the picnic table. Of course the fly (being relatively ignorant of your concerns) may choose to return to the scene, often from a new IP address. If this happens, it's time to pull out the big guns.

Step Five: Third and Final Warning

If you find yourself at this stage, it means that a killed abuser has returned to your Palace and shows signs of resuming the offensive behavior. The abuser may have simply waited until the death period expired, but in some cases the new attack will come from a different IP address — indicating that the abuser has access to The Palace via more than one computer. Once again, a warning is recommended; your intentions should be communicated in no uncertain terms. For instance: "Act up once more and you'll be banned from the site." At this point the smallest infraction should be enough to escalate the matter to the next step: we're not playing around anymore...

Step Six: BanIP/Ban

If the abuser is a member, the local gods can ban them by name. This is generally effective because the user's (Palace client's) regkey is used to prevent reconnecting under any name. If the abuser is a guest, however (as they often are), they may simply try to reconnect from somewhere else. Note that when abusive guests *do* move to another computer, it's usually within the same block of IP addresses. This indicates that the second computer is in close physical proximity to the first (they often belong to the same bank of machines located in a school computer lab, or fall within a single B block allotted for dynamic assignment by an ISP). Since in either case the abuser attains a new IP address, he or she is no longer affected by the commands you executed previously. This is why wizards have been given access to the *banip* command — in case of just such emergencies. This command not only allows you to ban an individual guest from connecting, but *all guests coming from the same block*. That means that until the ban expires or is lifted, no guest-level users from any of the affected IP addresses will be able to log in. Obviously this is a bit of "overkill," because everyone on the subnet is forced to suffer for the infractions of a single user, but you put it off as long as you could.

After the event

Abusers who have been denied connection for more than a few hours tend to stay away for a very long time, if not forever. What this really means is that after a few unsuccessful attempts to reconnect to your site, the abuser eventually gives up and goes somewhere else. Once in a while, however, you'll encounter a goofball with a lot of free time (and a lot of IP addresses), who decides that their mission in life is to give you a headache.

Fighting a maniac with a vendetta is a huge waste of time and energy, and is not recommended for the sane. Certainly the local wizards can ban the abuser whenever he or she shows up, but this isn't a satisfactory solution. You want to keep your cool, but you need a way to escalate your response. What should you do?

Since by this time the wizards have already run through the escalation procedure, followed all established security standards, banned the abuser and maintained logs of the event, they've done all they can do. The matter should now be handed over to the local Palace gods, who will render a decision on what to do next. Often a call or email to the administrator/owner of the offender's account can put things straight in a hurry; this is another good reason wizards should maintain a log for all such occurrences.

A perusal of the *god commands* list earlier in this document will show that only gods have access to the *ban* command (allowing them to ban a single IP address). This is because use of the ban command can be taken as an indication of "extreme prejudice" against a particular individual, and is therefore mitigated to the level of "systems administration" (in the traditional network sense). Beyond this point, matters should be handled between the administrators of the two sites involved (i.e., yours and the offender's). Most of the time, the administrators at the site of origin (the abuser's ISP, for example) will revoke the abuser's account privileges once incriminating or potentially controversial evidence is produced.

Authoring

Since the *Wizard Menu* and its attendant windows allow you to access and edit almost every line of code in the server script, authoring can be as simple or as complex as you want it to be. The gods of each Palace make their own rulings as to whether or not authoring will be allowed on the server, but in general it's not a good idea unless the wizards in question are very fluent in *Iptscrae*. Even then, it's usually a good idea to do any major revisions on another server, or during "closed" hours. This is because it's possible to inadvertently create a snag that will effectively "hang" the whole site, causing the Palace server to crash and disconnect all users online. When this happens, the server's prop file may become corrupted as well, creating even more work for the Palace gods. In addition, sudden major changes to the script will cause lag and may be very confusing to users.

That said, any wizard should be familiar with some of the basics of Iptscrae, if for no other reason than being able to diagnose a problem if no one else is around. The following subsections provide an overview of these scripting fundamentals.

Animation and scripting

With clever code and specially-designed graphics, it's possible to pull off a wide variety of animation effects in The Palace. The two basic types of animation possible within the Palace environment might best be referred to as *prop-based* and *spot-based*. No matter what effect you're looking for, however, before you can really make things "come to life" you'll need to an introduction to *Iptscrae*, the Palace programming language.

Say Ellohae to Iptscrae

Iptscrae (pig-latin for “script”) is a *Forth*-like, stack-based language used exclusively by The Palace software. The single most noticeable (and misunderstood) feature of the language is its syntax, which makes use of RPN (Reverse Polish Notation). Another term for RPN syntax is “postfix,” since the operators (or "verbs") in RPN are placed *after* the operands in a statement. This is in contrast to the “infix” method most westerners are familiar with, in which the operator is placed *between* its operands. If you’ve ever worked on a Hewlett-Packard programmable calculator, you’re familiar with RPN already. A few examples will illustrate the difference:

Infix:	Postfix:
2 + 3	2 3 +
(2 + 3) * 4	2 3 + 4 *
X = 4 - 3	4 3 - X =
printf("Howdy")	"Howdy" Say

Infix vs. Postfix Syntax

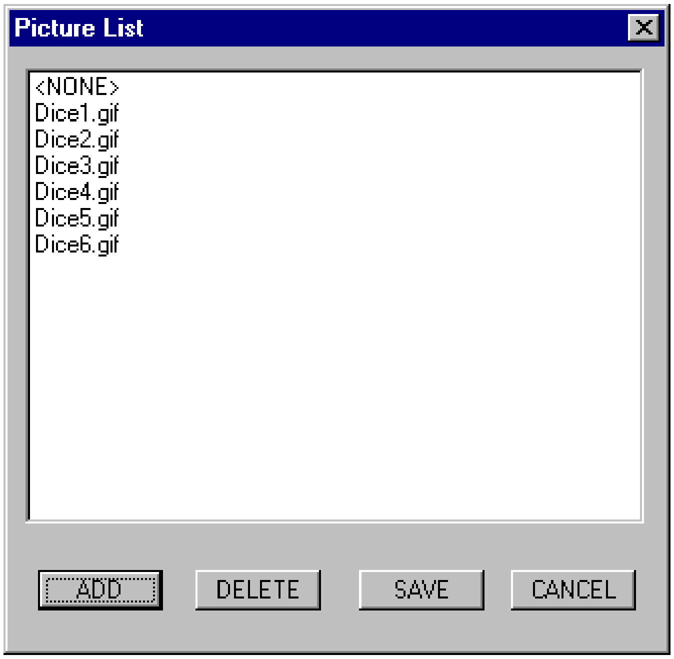
As you can see, the basic difference is that postfix word ordering places the effective “verb” (operator) after *both* “nouns” (operands): “two plus three” is phrased as “two three plus.” Despite how strange it looks to most of us, to a stack-based system like The Palace this method makes a lot more sense and wastes fewer system resources. The reasoning is simple: since your computer can’t add any two numbers until it knows what they are, there’s little reason to make it store the “plus” operator while you’re giving it the operands.

Of course, we can only *begin* to scratch the surface of Iptscrae’s potential here. As a fully-fledged programming language including over one hundred specialized commands and keywords, the very point of Iptscrae is to allow you to create new interactive scripts “from scratch,” and the only way to really get familiar with it is to *use* it. To give you something to start with, the following scripts illustrate a few fundamental animation effects. The first two examples cover the use of the picture list for both Windows and Macintosh users, and the third illustrates how to move a picture around dynamically. Don’t worry about the commands you don’t understand yet; you can always pick up a copy of *The Iptscrae Language Guide* online at the Palace website.

Example 1: "Tumblin' Die"

The example picture list shown here is used to create a randomized die in a hotspot. In its starting state (state 0), the die is not visible (since there is no graphic associated with the spot in this state, the background image is seen "through" it). States 1 through 6 correspond to the six GIF images used to represent possible rolls of the die. By clicking on the die (or more accurately, by clicking on the corresponding spot), the user triggers the following script:

```
ON SELECT {
 6 RANDOM 1 + dieRoll =
 dieRoll ME SETSPOTSTATE
}
```



Picture List

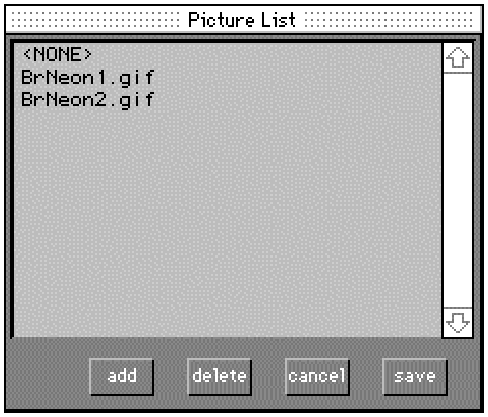
<NONE>
Dice1.gif
Dice2.gif
Dice3.gif
Dice4.gif
Dice5.gif
Dice6.gif

The Pictures List (Windows version)

Example 2: "Sign O the Times"

The picture list shown here corresponds to the flashing neon sign in "Harry’s Bar" (in the Mansion). In its starting state (state 0) no picture is displayed; the background image is seen. In state 1 the picture “BrNeon1.gif” is displayed, while state 2 displays “BrNeon2.gif”. Here's the script:

```
ON ALARM {
 {
 0 ME SETSPOTSTATELOCAL
 } {
  ME GETSPOTSTATE 1 + ME SETSPOTSTATELOCAL
 } ME GETSPOTSTATE 1 > IFELSE
 180 ME SETALARM
}
```



```
<NONE>
BrNeon1.gif
BrNeon2.gif
```

The Pictures List (Macintosh version)

Example 3: Picture-Moving

This script is a simple room-level demo that can be pasted directly into your script file. It causes a picture to move across the room and back again when clicked. Be sure to change the IDs and names so they apply to your server.

```
ROOM
   ID roomID
   NAME "roomName"
   PICT "pictureName.ext"
   PICTURE ID 100 NAME "picture.gif" ENDPICTURE
   SPOT
      ID 1
      NAME "Moving Picture"
      OUTLINE 100,200  250,200  250,260  100,260
; The “OUTLINE” assumes the picture is 150 by 60 pixels.
; Here come all the “states” the picture could be in...
; The first “state” is state 0, followed by 1, 2, etc.
; A picture’s “state” always consists of three things:
; pictureID (as above), horizontalOffset, and verticalOffset
; Notice that here we are only changing the horizontalOffset.
      PICTS 100,0,0 100,10,0 100,20,0 100,30,0 100,40,0 100,50,0
100,60,0 100,70,0 100,80,0 100,90,0 100,100,0 100,110,0 100,120,0
100,130,0 100,140,0 100,150,0 100,160,0 100,170,0 100,180,0 100,190,0
100,200,0 ENDPICTS
; Also note: that’s all one line, from “PICTS” to “ENDPICTS”
SCRIPT
ON SELECT {
   {
; Now we move the picture across the Room
; (from state 0 to state 20)
   { 1 ME SETSPOTSTATELOCAL } 1 ALARMEXEC
   { 2 ME SETSPOTSTATELOCAL } 10 ALARMEXEC
   { 3 ME SETSPOTSTATELOCAL } 20 ALARMEXEC
   { 4 ME SETSPOTSTATELOCAL } 30 ALARMEXEC
   { 5 ME SETSPOTSTATELOCAL } 40 ALARMEXEC
   { 6 ME SETSPOTSTATELOCAL } 50 ALARMEXEC
   { 7 ME SETSPOTSTATELOCAL } 60 ALARMEXEC
   { 8 ME SETSPOTSTATELOCAL } 70 ALARMEXEC
   { 9 ME SETSPOTSTATELOCAL } 80 ALARMEXEC
   { 10 ME SETSPOTSTATELOCAL } 90 ALARMEXEC
   { 11 ME SETSPOTSTATELOCAL } 100 ALARMEXEC
   { 12 ME SETSPOTSTATELOCAL } 110 ALARMEXEC
   { 13 ME SETSPOTSTATELOCAL } 120 ALARMEXEC
   { 14 ME SETSPOTSTATELOCAL } 130 ALARMEXEC
   { 15 ME SETSPOTSTATELOCAL } 140 ALARMEXEC
   { 16 ME SETSPOTSTATELOCAL } 150 ALARMEXEC
   { 17 ME SETSPOTSTATELOCAL } 160 ALARMEXEC
   { 18 ME SETSPOTSTATELOCAL } 170 ALARMEXEC
   { 19 ME SETSPOTSTATELOCAL } 180 ALARMEXEC
   { 20 ME SETSPOTSTATELOCAL } 190 ALARMEXEC
   } ME GETSPOTSTATE 0 == IF
   {
; ...and back to the start again
; (from state 20 to state 0)
   { 19 ME SETSPOTSTATELOCAL } 1 ALARMEXEC
   { 18 ME SETSPOTSTATELOCAL } 10 ALARMEXEC
   { 17 ME SETSPOTSTATELOCAL } 20 ALARMEXEC
   { 16 ME SETSPOTSTATELOCAL } 30 ALARMEXEC
   { 15 ME SETSPOTSTATELOCAL } 40 ALARMEXEC
   { 14 ME SETSPOTSTATELOCAL } 50 ALARMEXEC
   { 13 ME SETSPOTSTATELOCAL } 60 ALARMEXEC
   { 12 ME SETSPOTSTATELOCAL } 70 ALARMEXEC
   { 11 ME SETSPOTSTATELOCAL } 80 ALARMEXEC
   { 10 ME SETSPOTSTATELOCAL } 90 ALARMEXEC
   { 9 ME SETSPOTSTATELOCAL } 100 ALARMEXEC
   { 8 ME SETSPOTSTATELOCAL } 110 ALARMEXEC
   { 7 ME SETSPOTSTATELOCAL } 120 ALARMEXEC
   { 6 ME SETSPOTSTATELOCAL } 130 ALARMEXEC
   { 5 ME SETSPOTSTATELOCAL } 140 ALARMEXEC
   { 4 ME SETSPOTSTATELOCAL } 150 ALARMEXEC
   { 3 ME SETSPOTSTATELOCAL } 160 ALARMEXEC
   { 2 ME SETSPOTSTATELOCAL } 170 ALARMEXEC
   { 1 ME SETSPOTSTATELOCAL } 180 ALARMEXEC
   { 0 ME SETSPOTSTATELOCAL } 190 ALARMEXEC
   } ME GETSPOTSTATE 20 == IF
} ENDSCRIPT
ENDSPOT
ENDROOM
```

5 Palace Design Tips & Techniques

This section describes various means of adding advanced functionality to your Palace site, and includes an examination of the limitations faced by Palace designers.

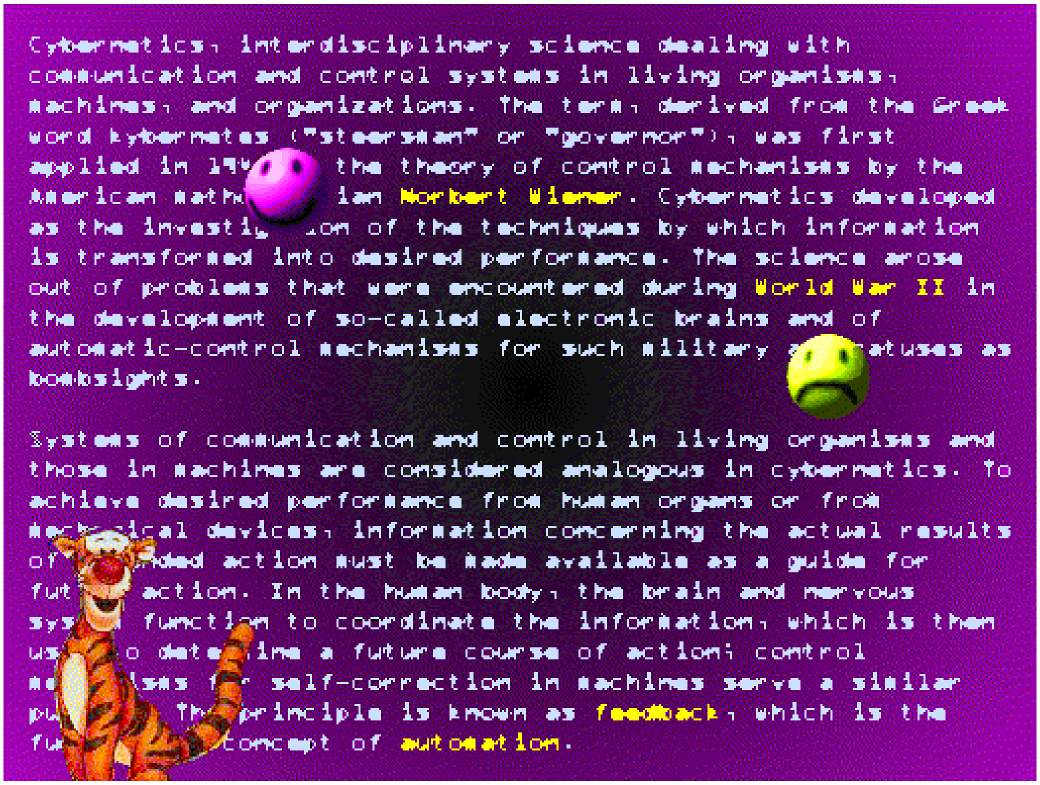
- *Linking Palace sites and web sites* on page 63
- *Palace limitations and workarounds* on page 68

Linking Palace sites and web sites

In the area of “DTVR” (DeskTop Virtual Reality), the web clearly reigns supreme as the medium of choice. Indeed, with clever use of some advanced HTML tags (and “enhancements” to the HTML specification), web pages become “dynamic documents,” capable of not only output functions, but of receiving input as well. With the addition of forms, plug-ins, media streaming, VRML and distributed multimedia applications, the web is becoming more immersive and interactive all the time. These new technologies are helping to redefine the functional model of the web, moving it further and further from its roots in traditional layup and DTP. This is good news indeed for webmasters.

Nevertheless, anyone who’s attempted to create a true feeling of *environment* on the web will tell you that this is still a severely daunting task, the results of which are more often unsatisfying than successful. VRML, although exciting and promising, has yet to make it out of “public beta,” and is still a non-option for that great percentage of users who browse the web using older computers and 14.4Kbaud connections. When all is said and done, web pages are still *pages*, and this is not likely to change real soon. For the time being (at least), this indicates a key functional limitation of the web proper.

In contrast, the functionality of The Palace assumes an *environmental* design model, rather than a *publishing* one. Of course each approach has its strong points, but only when utilized in the right type of situation. For example, it is possible to use Palace rooms as “pages” (by creating background graphics which are full of text), but the cost in bandwidth and the limited area of the *View Screen* make this a pretty inappropriate use of the medium…



Cybernetics, interdisciplinary science dealing with
communication and control systems in living organisms,
machines, and organizations. The term, derived from the Greek
word kybernetes ("steersman" or "governor"), was first
applied in 194 the theory of control mechanisms by the
American math ian Norbert Wiener. Cybernetics developed
as the investig ion of the techniques by which information
is transformed into desired performance. The science arose
out of problems that were encountered during World War II in
the development of so-called electronic brains and of
automatic-control mechanisms for such military a atuses as
bombsights.

Systems of communication and control in living organisms and
those in machines are considered analogous in cybernetics. To
achieve desired performance from human organs or from
mech ical devices, information concerning the actual results
of nded action must be made available as a guide for
fut action. In the human body, the brain and nervous
sys function to coordinate the information, which is then
us o det ine a future course of action; control
me isms f r self-correction in machines serve a similar
pu The principle is known as feedback, which is the
fu concept of automation.

“CyberDictionary” Room, Imaginary (Terrible) Design

If this design approach is carried to an extreme (essentially creating Palace-based hypertext), the “multiple user problem” manifests on another, more obnoxious level. Looking at the illustration above, it’s all too easy to imagine the following dialogue:

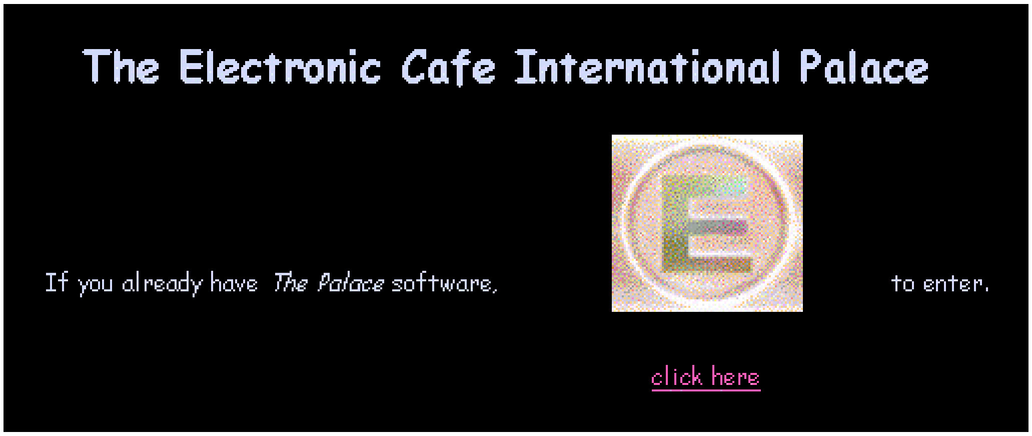
```
USER 1: Could you move, please?  I’m trying to read the first paragraph.

USER 2: Yeah, tell that to Tigger – hey TIGGER!  Move your avatar!
People are trying to READ here!!!
```

Tigger’s user, meanwhile, is busy on another line, or has gone into the kitchen to grab some cookies and milk. Perhaps he’s even fallen asleep in his chair! The lesson here is very simple, but it bears repeating: just because you *can* do something, that doesn’t mean that you *should*. To see how such decisions might impact upon your own design process, we will next take a look at a less fanciful scenario, this time taken from a real life experience.

HyperLinks: From Web to Palace

In HTML, creating a means of travel from your web page to your Palace site is as simple as making any other sort of link; you simply insert an HREF tag. For example, the big "E" graphic on the Electronic Café International's "palaceinfo" web page also functions as a hyperlink to the ECI Palace itself.



The Electronic Cafe International Palace

If you already have *The Palace* software, to enter.

click here

The HTML for the "Big E" graphic looks like this:

```
<a href="palace://ecipalace.metawire.com"><img src="BigE.gif"></a>
```

As you can see, the only unusual element is the "palace://" identifier, which the web browser understands to mean "Launch The Palace client to display this URL." When web users click on the graphic (or text) in question, The Palace client is shelled (i.e., executed by, or "through," another program). The browser window will drop to the background as The Palace client becomes the active window (the browser continues to run in the background while The Palace client is running, and may still be used as usual). When the user is finished exploring your Palace, simply exiting The Palace client will return them to your web page, and the browser becomes the active window again.

HyperDoors: From Palace to Web

The creation of a "hyperdoor" from your Palace to your web page is no less straightforward, although (of course) the code is written in *Iptscrae* rather than HTML. Only the wizards or gods of the Palace in question can create hyperdoors:

▼ To create a hyperdoor

1. Select **New Door** from the **Wizard** Menu (or select an existing door while in *Authoring Mode*).
2. Select **Door Info…** from the **Wizard** Menu (or double-click on the door itself) to bring up the **Door Info** window as described in *Door Info…* on page 22.
3. Click on the **Type** combo box and select **Normal**.

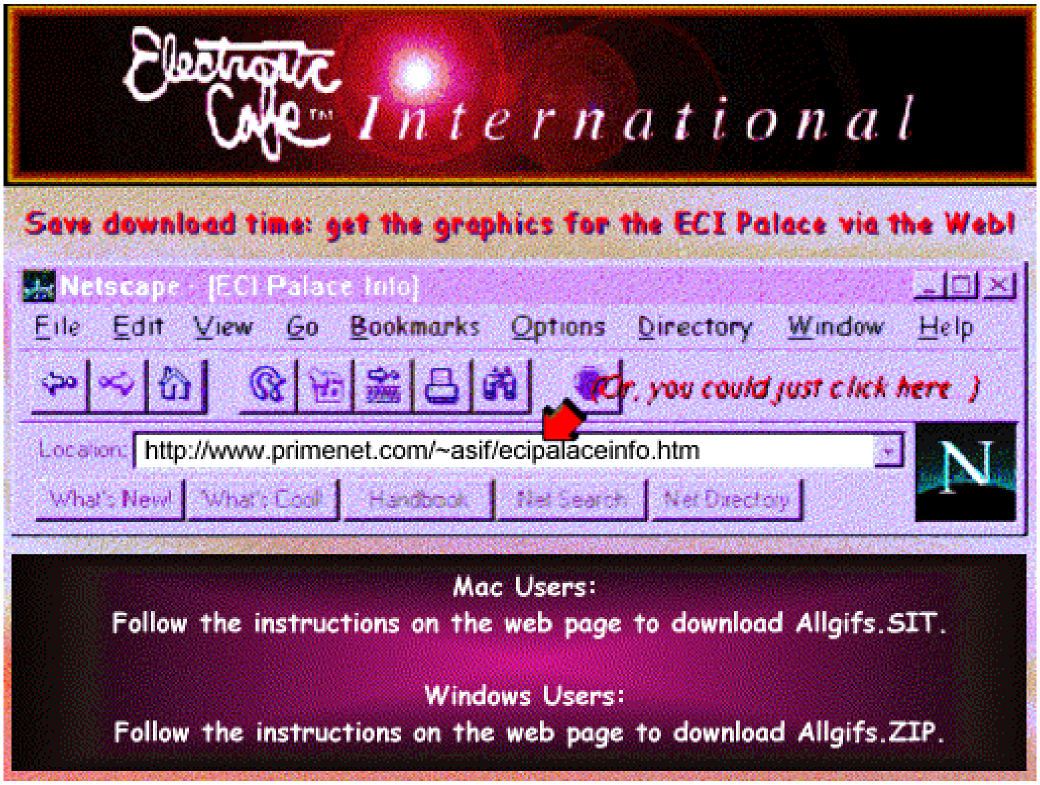
4. Click on the **Script** button to bring up the Scripting window.
5. Insert the following line of code and you're done:

```
ON SELECT {
"http://<your_target_URL>" NETGOTO
}
```

NOTE – You can also create doors between Palaces using this same command. If the URL specified begins with a "palace://" identifier, the door will lead to the indicated Palace site.

Case History: The Good, the Bad, and the Totally Stupid

In the original design for the *ECI Palace* a door was placed in the entrance room, labeled "Click here for info on downloading the graphics." Clicking on this door sent the user to the room shown below…


Electronic Cafe™ International
Netscape - [ECI Palace Info]
File Edit View Go Bookmarks Options Directory Window Help
Or, you could just click here)
Location: http://www.primenet.com/~asif/ecipalaceinfo.htm
What's New! What's Cool! Handbook Net Search Net Directory
N

Save download time: get the graphics for the ECI Palace via the Web!

Mac Users:
Follow the instructions on the web page to download Allgifs.SIT.

Windows Users:
Follow the instructions on the web page to download Allgifs.ZIP.

"Get the Graphics" room, Original (Poor) Design

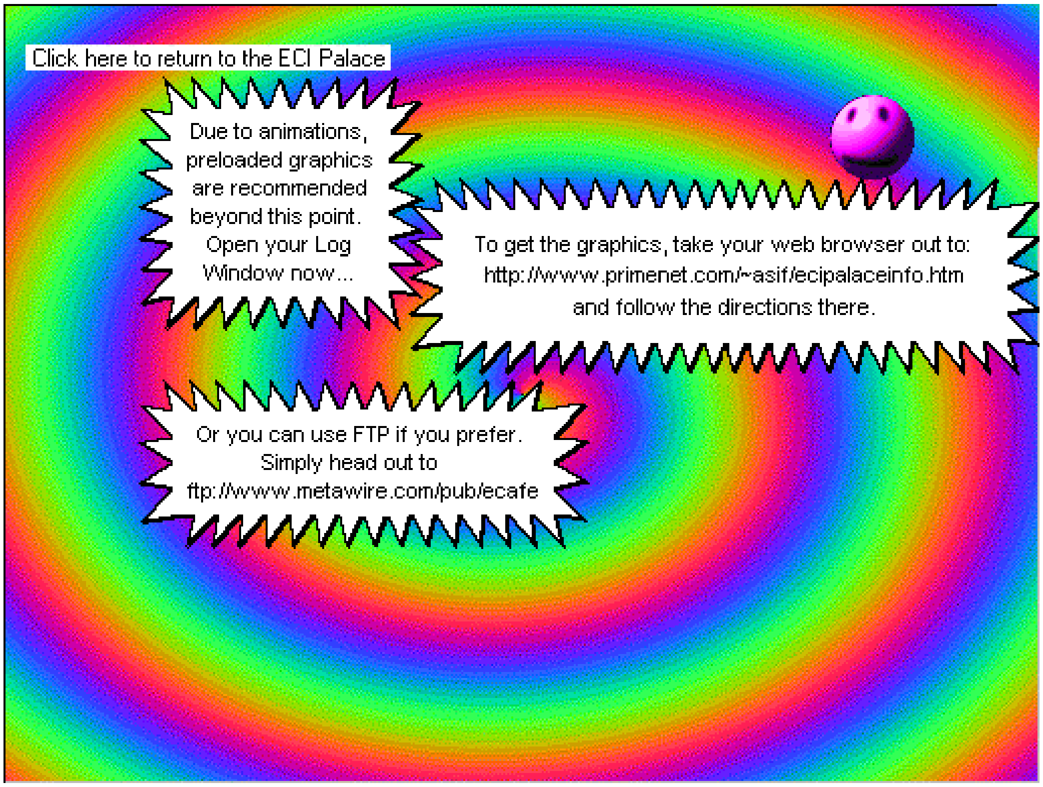
This design seemed like a good idea at the time: it was informative, goal-directed, and fairly neat to look at, but it turns out that users didn't appreciate it much at all. In fact, new users often *left* this room to go ask a wizard for all the same information. Obviously, something needed fixing. Care to guess?

The problems with this design are manifold, but are all related to its reliance on graphically-depicted text in the background layer. The instructions are literally *part* of the background image, which of course opens us up to the "blocking" phenomena described above, but the problems don't end there. Here are a few more: (1) Since they're part of the graphic, the instructions won't be seen until the picture makes it down to the user’s hard disk. (2) Since the art is *original* (i.e., not part of any other Palace site users may have been to), it needs to be sent to practically every user who comes in. (3) This room, unlike most, is almost certain to be avoided by those who’ve been there once before; this further decreases the value of having to deal with it at all. (4) The intensely-dithered nature of the graphic makes for a larger file, causing the download to take longer than it should for such a utilitarian room. There are probably other problems, but you get the idea.

All these little problems add up to one really big one. Simply put, new users would come here to get "quick information" and instead ended up watching a blank screen while they waited for a large piece of art to download. Worse still: if several new users all entered the room at the same time, the sudden rush of server activity would spawn a timesharing cycle (a “daisy-chain of lag”), further slowing an already inefficient process.

The second version of this room addresses the problems raised by its predecessor. Looking at the illustration below, you can immediately see that some drastic changes have occurred. In fact, the current version of the room bears almost no resemblance – visually or mechanically – to the original. Let’s examine what was done to fix this troubled design. There were three main elements to the solution.

- First, the original graphic was replaced with one of the pictures from the "Mansion" Palace (“Spiral.GIF”). Since so many people already have a copy of this image, downloading is often not required.
- Second, the instructions were moved into chat balloons (of the “excited” variety). These have the admirable quality of showing up on the screen whether or not the background picture is in place yet.
- Finally, the "spoken" instructions now tell the user to open the *Log window*, where (as you might expect) the words from the chat balloons appear once more. Many users (especially new ones) are unaware of how useful the Log window is; this approach allows them to select an address right out of the log, copy it, and paste it into their browser’s "Location" box.



Click here to return to the ECI Palace

Due to animations, preloaded graphics are recommended beyond this point. Open your Log Window now...

To get the graphics, take your web browser out to: http://www.primenet.com/~asif/ecipalaceinfo.htm and follow the directions there.

Or you can use FTP if you prefer. Simply head out to ftp://www.metawire.com/pub/ecafe

“Get the Graphics” Room, Second Design

Since we could give users the ability to choose their own preferred downloading method (browser or FTP), this represented an ideal solution; if you cannot provide FTP access, a browser download will be just fine by itself. Of course, as you have already learned, it is also possible to create direct *hyperlinks* and *hyperdoors* between Palace rooms and web pages. These are very helpful things for users to have access to, providing them with an easy, familiar way to obtain your graphics and sounds.

However, this is only the most obvious of many possible uses for such links. In fact, Palace rooms and web pages can be interwoven so easily that it becomes quite natural to begin imagining *hybrid sites*, in which The Palace client and the web browser play equally important roles. Building a network of interconnected Palace rooms and web pages allows you to do all sorts of interesting things that neither The Palace nor HTML could support alone.

Palace limitations and workarounds

The creation of a basic Palace with individual rooms, doors, and props is a fairly simple project, comparable to the creation of a typical web page (without CGI, Frames, or other advanced functionality). After you’ve got your first Palace up and running, however, you’ll probably start imagining ways to make it more immersive, complicated, interactive, and computationally intense. At this point, you’ll probably also begin bumping into some of the Palace’s hard-wired limitations, which are rigid enough to warrant a special section of their own.

In the pages that follow we will look at some of these limitations, and discuss some of the ways Palace designers have found to work around them. We will also look at the advanced functionality currently being offered by progressive Palace sites, and discover how they were accomplished.

File sizes and preloading

One area of concern for the Palace developer is the sheer size of the images used to depict the rooms. These images are always 512 by 384 pixels, and are generally saved as 8-bit (256-color) GIFs. Since GIF images are generally 25% to 50% as large as flat bitmaps (in terms of bytes required for storage), we can determine the typical size of these graphics by applying the rule of thumb for bitmaps and dividing that number by sigma (the range of) two to four:

1. Bitmap Size Determination

   (512 pixels * 384 pixels * 8 color bits) / 8 = 196608 bytes

2. GIF Translation:

   196608 / 4 = 49152 bytes (low end)

   196608 / 2 = 98304 bytes (high end)

As you can see, the average Palace GIF will come in between 50 and 100 kilobytes. This is not a “problem” as far as the software is concerned; if a user doesn’t already possess the graphics for a room they have entered, the server simply sends them the required files. This has a few disparaging effects, however. First, it forces the user to wait for one to two minutes every time she enters a room she’s never seen before. Just as on a web page, this waiting process frustrates many users -– especially those with slower connections — and may end up negatively affecting their opinion of the site as a whole. Second, the sending of these images is yet another task the server must tend to in realtime; this causes all other server tasks – like animations in other rooms, etc — to “timeshare,” and goes against one of the general design tenets of The Palace, which is to do everything possible *on the client’s side of the connection*.

These are really more like “facts of life” than “problems,” but there is one more consideration regarding the “timesharing” process explained above: The Palace has no built-in way of determining when a download has ended. You can’t write code that uses a “download completed” signal as a trigger of any kind. In complex auto-activated rooms (rooms containing constantly-looping animation routines or the cyclic triggering of alarms, for instance), the Palace script will dutifully continue flipping graphics *even though the user isn’t seeing any of this activity yet*. In effect, the server’s “timesharing” approach works *directly against* the downloading user, dragging out an already-undesirable process.

If your site uses a lot of "spot art" or animations, here's how you can minimize this problem right now:

1. Compress all your original Palace graphics (and sounds) into a single file, using a common archive format such as .SIT or .ZIP. You'll probably want to do one for each platform (SIT files are created by *Stuffit*, while ZIP files are created by *PKZIP* or *WinZIP*).
2. Make these files available via your web page or anonymous FTP site.
3. Place a message or script just inside the “front door” of your Palace, informing people where they can go (usually via the web) to "preload" the graphics and sounds. For ease of use, it’s a good idea to build a “hyperdoor” to your archive’s URL right there in the same room. This is easily done – requiring only one line of code — and very effective. When the user clicks in the hyperdoor (which is labeled something like "Click here to get the graphics"), The Palace client automatically launches the web browser and heads out to the URL specified in its script. Once there, the user can download your graphics and sounds simply by clicking on the appropriate link.

Conversely, it’s also a good idea to provide a link *into* your Palace *from* your web page. Until The Palace becomes as common as the web itself — hey, it could happen — it’s safe to assume that most people will come across your web page first. While you’re at it, you might want to provide a link to The Palace Home Page (http://www.thepalace.com), where users can download the software and find out how it works.

Art and Palette Limitations

Speaking of graphics, there are a few more limitations you ought to be aware of before you dive in and start creating images for backgrounds and animations…

The Palace Palette

As mentioned above, all Palace artwork — room backgrounds, midground pictures, avatars and props -– is done in 256 colors. *The same* 256 colors. To allow fast, platform-independent color fidelity, The Palace makes use of a special palette called the “M&M Palette,” which contains three intergradiated ramps of red, green and blue values, a fourth ramp of white-grey-black values, and the twenty colors of the *windows* system. (The name comes from the passing resemblance the palette bears to a bunch of square M&M candies.)

Despite the possible creative frustration felt by 24-bit-color purists, the M&M palette is a very good one; you can use it for all sorts of stuff besides Palace design, it works very well for color reduction (from 24- or 16-bit to 8-bit).

Since it exists in every piece of Palace art, the M&M Palette is easy to obtain; simply log into any Palace Site and download one picture. After that, use a quality graphics editor like Adobe's *Photoshop* or JASC Inc.'s *Paintshop Pro* to extract the palette from this picture (you’ll find the downloaded GIF sitting in your *Pictures* folder, which is located by default inside your *Palace* folder).

To avoid unexpected color performance and extra work, be sure to apply the M&M palette to your new images before you go too far with them.

Maximum Pixels per Room

When you begin designing room-based animations in your Palace, you may run into another limitation: there is a maximum on the amount of graphic data that can be associated with a room. This limit is hard-coded into the system, to ensure that users on slower computers don't get completely bogged down. Without getting into the technical details of CRC-checks and other related factors, this limit is roughly three or four times the total graphic area of the *View Screen*, or up to 1024 by 768 pixels. (Note that the Windows version of The Palace client can display more graphic area than the Macintosh version, but unless you're building a Windows-only environment, it's unwise to go past this limit. This also implies that when testing for maximum pixels per room, the Macintosh version should be used.)

1024 by 768 pixels is a nice amount of space to play with; quite adequate for your typical exploratory environment (e.g. a "Myst"-like experience). If you can live within this limit, you can do some pretty cool stuff. For instance, you could do an 8-frame animation filling half the *View Screen* — in one room. Or you could create an animated character about the size of two props (44 by 88 pixels) and give it 32 possible positions!

The best defense against running out of graphic space is cleverly designing individual elements of your art so that they cooperatively display, hide and reveal what you want to, using the smallest number of pixels possible.

Props

It should be borne in mind that props — new ones, especially — are the single largest cause of network traffic related to a Palace site. This is because the more users there are in a room, the more people need to receive graphic information every time you don a prop they haven't seen before. If your users are real clothes-horses (or "prop-horses" if you will), all of this prop activity may create a noticeable amount of lag.

If your Palace site is lagging badly, you can try this simple experiment to determine if prop traffic is to blame: simply turn custom props off on the server (you should warn your users before doing this). If the lag decreases while carrying a normal amount of users, props are the problem.

Since you probably don't want to leave the custom props flag off indefinitely, there's little you can do about this situation (short of getting more bandwidth, closer to the Internet backbone). Of course since your users are the people most affected by the lag, it may help to explain the cause of the problem to them and ask if they could let up on the new props a little. Who knows? They may even cooperate!

Audio limitations

As of this writing, The Palace software supports use of .WAV files for both Macintosh and Windows platforms, and .MID (MIDI) files for Windows only. The sound files which come with The Palace software –- they’re in the WAV format, even though they have no file extensions — were sampled at 22 kHz in 8-bit mono. The Windows version of The Palace client can play WAV files at higher sampling rates and greater bit depth, but the cost in terms of download time is usually not worth it. In addition (and much more importantly), the Macintosh version will not be able to play these files: they will be heard only as terrible screeching sounds.

Like graphics, any new sounds you create should be made available via web-based hyperlink or FTP site, as explained in *File sizes and preloading* on page 69. Here’s how to prepare them:

1. Mix and edit your sounds down to the lowest fidelity you can (without utterly destroying the sound quality).
2. Save them as WAV files.
3. Change their names to remove the `.wav` extensions.
4. Compress all your soundfiles together into a single archive file, such as “ALLSOUND.ZIP” or “AllMyWaveFiles.SIT.” It’s a good idea to do an archive for each platform (SIT files are created by *Stuffit*, while ZIP files are created by *PKZIP* or *WinZIP*).
5. Make your compressed archives available via your web page or FTP site, just as you did with your graphics. You might decide to compress both sound and graphics into one massive file.

Code limitations

No matter how cleverly you assign variable names and swap graphic elements, there are some datahandling limitations hard-coded into the software that you will not be able to change. Here are the most important limitations to keep in mind:

- The maximum length of a string is 4096 characters.
- The maximum number of local variables (in a given room) is 64.
- The maximum recursion (number of nested subroutines) is 16.
- The maximum number of alarms that can be queued at one time is 32.
- The maximum number of data items that can be held on the stack is 32.
- The maximum number of states that can be assigned to a spot is 32.
- The maximum number of elements that can be assigned to an array is 32, although you can get around this by storing arrays within arrays.

Of all the above limitations, the one you'll really want to look out for is the *Stack Limit* — if more than 32 data items get placed on the stack, some of them will “fall off,” becoming lost forever. In general, however, these code limitations really shouldn't be too much of a problem for you (unless the goal is to do truly complex stuff like tracking the states of users and objects from room to room, for instance). This is left as an exercise for the reader.

A The ~club Command

~club

Like the special commands described earlier, this command begins with a tilde (~) rather than a quotemark. The *~club* command is designed for use at public clubs and cafes, where The Palace might be running unattended for long periods of time. The *~club* command disables the following features:

- Display of Registration Number on the splash screen
- Child Lock
- Launching of Web Browsers
- Links to other Palace sites

B Palace Command Summary

The following table provides a list of all special commands (not counting Iptscrae commands and keywords):

- *Guest commands* on page 75
- *Member commands* on page 75
- *Wizard commands* on page 77
- *The god commands* on page 78

Guest commands

The following table lists the Palace guest commands.

Guest Commands	Syntax	Description
help	`help	Displays all special commands available to your user class, in the Log window
page	~page *message*	Summons help from a wizard

Member commands

The following table lists the Palace member commands.

Member Commands	Syntax	Description
add	'add *username*	Sends an invitation to the user to join your buddy list.
hide	`hide on `hide off	Hides you from all other users Unhides you from all other users
hidefrom	`hidefrom *name*	Hides you from user *name*
mute	`mute *name*	Mutes user *name*
newroom	`newroom `newroom [*roomName*]	Creates your member room (one per user) Creates your member room as *roomName*
locate	*locate *buddyname*	Locates a buddy on the user's buddy list.
page	`page *message* ~page *message*	Summons help from a wizard
page	*page *buddyname* "*message*"	Sends a message to a buddy on the user's buddy list.
password	`password *password*	Specifies a password for navigation purposes
rclose	`rclose	Closes your member room to further visitors
rdelete	`rdelete	Deletes your member room
re respond	`re *message* `respond *message*	Replies to the last sender of an ESP message
rejectesp	`rejectesp on `rejectesp off	Rejects all incoming ESP messages Stops rejecting ESP messages
rejectprivate	`rejectprivate on `rejectprivate off	Rejects all incoming private messages Stop rejecting private messages
rguests	`rguests on `rguests off	Allows guests to enter your member room Prohibits guests from entering your member room
rhide	`rhide on `rhide off	Hides your member room from the Room List Unhides your member room from the Room List
rkick	`rkick *name*	Kicks *name* out of your member room
rname	`r*name roomName*	Renames your member room as *roomName*
ropen	`ropen	Opens your member room to visitors
rowner	`rowner *name*	Transfers ownership of your member room to user *name*
rpainting	`rpainting on `rpainting off	Enables painting in your member room Disables painting in your member room
rpassword	`rpassword [*password*]	Password-protects your member room
rpicture	`rpicture *filename*	Sets new background for your member room

Member Commands	Syntax	Description
rscripts	\`rscripts on \`rscripts off	Enables scripts in your member room Disables scripts in your member room
runhide	\`runhide	Same as \`rhide off
runkick	\`runkick *name*	Cancels an \`rkick command
unhide	\`unhide	Same as \`hide off
unhidefrom	\`unhidefrom *name*	Unhides you from user *name*
unmute	\`unmute *name*	Unmutes user *name*

Wizard commands

The following table lists the Palace wizard commands.

Wizard Commands	Syntax	Description
autoannounce	\`autoannounce *message*	Sets the server greeting to *message*
banip	\`banip [*minutes*] *ipAddress*	Bans one or more IP addresses for *minutes*. May use wildcards (i.e.: 12.24.23.*).
banlist	\`banlist [-k] [-a] [*name* | *ipAddress*]	Shows banned users in log window
banuser	\`banuser [*minutes*] *name*	Bans user *name* for *minutes*
bots	\`bots on \`bots off	Turn cyborgs on server-wide Turn cyborgs off server-wide
comment	\`comment *ipAddress message*	Adds a comment to a banlist
death	\`death *minutes*	Set default death penalty to *minutes*
delete	\`delete	Deletes the current room
duplicate	\`duplicate	Duplicates the current room
er	\`er	Moves you to the room of the last (non-wizard) page
extend	\`extend *string minutes*	Extends the death penalty on a ban record
flood	\`flood *number*	Kill flooders after *number* events per second
gag	\`gag *name*	Gags user *name*

Wizard Commands	Syntax	Description
glist	`glist [-n | -k | -l]	Lists information on all users
kill	`kill [*minutes*] *name*	Kills (disconnects) user *name* for *minutes*
list	`list [-n | -k | -l] [*name* | *ipAddress*]	Lists information on room users
paint	`paint on `paint off	Turns painting on server-wide Turns painting off server-wide
pin	`pin *name*	Pins user *name*
propgag	`propgag *name*	Propgags member *name*
purgebanlist	`purgebanlist	Purges elapsed ban records
purgeprops	`purgeprops *number*	Purges all props older than *number* days
repage	`repage *message*	Sends *message* to the last (non-wizard) page
roommaxguests	`roommaxguests *number*	Sets the max guest occupancy for the current room
roommaxocc	`roommaxocc *number*	Sets the max occupancy for the current room
sortprops	`sortprops	Sorts the server's prop file index
track	`track [*minutes*] *name*	Tracks all sign-ons from user *name* for *minutes*
trackip	`trackip [*minutes*] *ipAddress*	Tracks all sign-ons from *ipAddress.* May use wildcards (i.e.: 12.24.23.*).
unban	`unban *name* `unban *ipAddress*	Unbans user *name* Unbans user from ipAddress
ungag	`ungag *name*	Ungags user *name*
unpin	`unpin *name*	Unpins user *name*
unpropgag	`unpropgag *name*	Unpropgags member *name*
untrack	`untrack *name* `untrack *ipAddress*	Stops tracking user *name* Stops tracking user from *ipAddress*

The god commands

The following table lists the Palace god commands.

God Commands	Syntax	Description
authoring	\`authoring on \`authoring off	Allows wizards to enter Authoring Mode Prevents wizards from entering Authoring Mode
ban	\`ban *name*	Permanently banishes *name* from the server
botkill	\`botkill on \`botkill off	Allows cyborg scripts to kill Prevents cyborg scripts from killing
chatlog	\`chatlog on \`chatlog off [*filename*]	Enables chat logging on the server Disables chat logging on the server
custom	\`custom on \`custom off	Enables custom props on the server Disables custom props on the server
defaultroomocc	\`defaultroomocc *number*	Sets the default room occupancy
downlist	\`downlist *number*	Moves the current room *number* positions down in the script file
dropzone	\`dropzone	Makes the current room a dropzone (front gate)
godpass	\`godpass *password*	Sets the god password to *password*
guests	\`guests on \`guests off	Enables guest access to the server Prevents guest access to the server
killprop	\`killprop *propID*	Kills a specific prop ID
maxocc	\`maxocc *number*	Sets the maximum server occupancy
memberrooms	\`memberrooms on \`memberrooms off	Allows members to create member rooms Prohibits creation of member rooms
nowhisper	\`nowhisper on \`nowhisper off	Disables whispering Enables whispering
passwordsecurity	\`passwordsecurity	Enables password security (auto-kills after three failed password attempts)
picdir	\`picdir *path*	Sets the source directory for outbound files to *path*
playerkill	\`playerkill on \`playerkill off	Allows members to kill Prohibits killing by members
purgelimit	\`purge *number* \`purgelimit *number*	Sets the purge limit to *number* days Same as \`purge *number*
recycle	\`recycle [*number*]	Causes guest ID numbers to recycle at *number*
server*name*	\`servername *string*	Sets the servername to *string*
shutdown	\`shutdown	Shuts down the server
spoof	\`spoof on \`spoof off	Allows members to spoof using @x,y commands Prohibits spoofing

God Commands	Syntax	Description
uplist	`uplist *number*	Moves the current room *number* positions up in the script file
wizardkill	`wizardkill on `wizardkill off	Allows wizards to kill Prohibits killing by wizards
wizards	`wizards on `wizards off	Enables support for wizards Disables support for wizards
wizardsonly	`wizardsonly on `wizardsonly off	Sets the current room as wizard only Removes wizard-only setting
wizpass	`wizpass *password*	Sets the wizard password to *password*

Index