

City of Heroes/City of Villains Keybind, Macro & Emote Guide

V2.00 – April 2009 (Issue 14)

Compiled by
The Legendary Death-Defying
Shenanigunner

@Shenanigunner or shenanigunner [at] dgath [dot] com
aka (on Victory) [Sgt Glory B](#), [Rorseach](#), [Mr Cellophane](#), [R A Heinlein](#), [BrotherJustin](#), [Sir Richard F Burton](#),
[Chain Chain Cheney](#), [V-Enetta Count Q](#).
aka (on Virtue) [Doctor Gigglez](#), [Sioux Sheff](#), [Vond](#)
Now always found at <http://tinyurl.com/cohbinds>

TOC

Section 0	Introduction
Section 1	Basics
Section 2	Keybinds
Section 3	Macros
Appendix A	Complete List of Console Slash Commands
Appendix B	Group List of Console Slash Commands
Appendix C	Complete List of Bindable Key Names
Appendix D	Complete List of UI Window Names
Appendix E	Complete List of Emote Codes
Appendix F	Chat Bubble Color Controls
Appendix G	Saving and Loading Interface Settings
	Revision History

SECTION 0 – INTRODUCTION

0.1 What are Keybinds & Macros?

Keybinds and macros are ways to remap the keys, mouse buttons and game commands into a control configuration that better suits a given player's play style and preferences. Instead of being locked into a fixed set of control keys and commands, as some older games do, or providing a simple reassignment feature as most newer games do, sophisticated games like City of Heroes/City of Villains permit you to remap, change and combine the game commands and controls in an almost unlimited fashion.

For example, instead of simply letting you change your "run" command from the default R key to another, keybinds allow you to bundle two or more commands onto one key, so that you initiate running and go into Super Speed at the same time. Another example is the very useful "engage" keybind, which targets the nearest foe and locks you onto him in "follow" mode. For a melee player (scrapper or tanker) in the middle of a multi-foe fight, being able to whack one key and lock onto a foe for focused attacks can change your play style and success rate.

Or, on the fun and silly side, you can combine chat strings with actions – a local string "C'mere, you ugly SOB!" with a taunt, or "Roast in hell!" combined with a AoE (multi-foe area of effect, that is) Burn or Scorch power. Or "Let's get 'em!" combined with a suitable emote (character animation), to tell your teammates it's time to get down to business. (There are several of those predefined in the QuickChat emote menu.)

A macro is exactly the same as a keybind, except that the string of commands is bound to a power-tray button and has to be activated by a click or an associated power-activation command (by default, the associated keyboard-top number or alt-number). Generally, you should use keybinds for commands and command sequences you need to activate quickly and often, while macros can be used for actions for which you'll have time to find and click a power button.

Some players might be happy with a few reassignments from the default command keys. Others might want to do the crazy thing and create a completely custom mapping of everything. However, all players can benefit from a few keybinds that make key powers speedier and easier to use in the heat of battle.

0.2 Why do I need a Keybind & Macro Guide?

Creating effective keybinds (or binds, for short) and macros takes some knowledge and experience, which the basic game manuals don't really cover. At a minimum, you need to know the basic syntax for writing a bind or macro, and have a list of all the console or "slash" commands (so called because they begin with a slash that identifies them as commands when you type them into the chat window).

Since the developers of CoH don't provide a comprehensive guide, and are a little loose about providing consistent information about the list of available slash commands with each update, it's fallen to the player community to keep track of the commands and teach each other how to use them. A guide of some sort is essential to help you master this complicated and flexible set of commands.

0.3 There doesn't seem to be any shortage of Bind & Macro Guides. Why this one?

Any time documentation is written by a community of users, you're going to see some common limitations. Those who are real hotdogs with the tools may not be very good at writing about them; those who can write well may not know enough about the process to get all the details right; even those who can do both might not have time to gather all the details or keep things updated.

I set out to write this guide because, as a player new to CoH (as was everyone at one time) and a player new to multiplayer online games, I couldn't find a good, complete, up-to-date guide that was written in language a non-MMOG maven could understand. The guides that were any good in the information department seemed to be written in poorly-translated Martian, assuming far too much previous knowledge on the part of the reader. And the guides that were incomplete, sloppily compiled and out of date were even more frustrating for a newcomer, because it was hard to identify what was right and what was useless.

So here's this guide, added to the pile, and I hope an improvement and of value to both new and existing players. My aim was to combine many years of experience writing software and programming manuals (most for novice and nonexpert users) with my fascination of City of Heroes and all the reliable information I could lay hands on. It's my aim to keep it updated, both with new info gleaned from the forums and other users, and from reader feedback.

0.4 How This Guide is Organized

The organization of this guide is simple. Section 0 is the Introduction, which you're reading. Section 1 is Basics, like terminology and syntax, which I urge all users to read carefully, so that they can follow the terser language in the following sections. Section 2 explains how to write and use Keybinds. Section 3 covers Macros (mostly, in how they differ from Keybinds). Appendix A lists all the known console or slash commands, with their individual syntax and notes on how to use them. Appendix C lists the names of all the bindable keys. Appendix D lists the known window and menu names. And Appendix E lists all the known "emote" commands, with notes where appropriate.

0.5 Updates

For many reasons, this will probably be the final update of this guide. The last update was around Issue 8 in late 2006 and not very much grew out of date; even with massive changes to the game the command and emote set hasn't changed as much as in prior Issues. I am also nearing my end of involvement with the game, at least to the point where I am inspired to maintain this guide. But you never know...

Updates and corrections, especially to the command and emote lists, are encouraged. Comments on everything are welcomed. And pestering when I let the guide fall out of currency is solicited – I have a tendency to move on and not maintain efforts like this, especially when there's no feedback.

Email to **shenanigunner [at] dgath [dotcom]**, or to whatever maintenance email address is listed on the web site at **<http://www.dgath.com/coh/>**, is the best way. You can also send in-game mail to **@Shenanigunner**, or tap me whenever I'm online. My global chat handle is **@Shenanigunner**.

0.6 Acknowledgements

Only the general presentation of this guide, along with much direct verification of the commands, is solely mine. All of the information came from other sources – mainly, the game itself, its user manual, and the Prima game guide.

The original list of slash commands was provided by Xocyll, in the Usenet forum alt.games.coh, copied from the Binds forum on the official CoH web site. Xocyll has also posted a number of discoveries of his own on the Usenet group, which are included here, and has provided continual feedback on the guide. Neil Cerruti provided some useful info and feedback as well for the I6/I7 material.

A lot of the basics came from other guides and helpful people in the forums. As nearly all of it traces back to information from the game developers, and because I didn't keep track of who told me what, all I can do is offer a general, generous and heartfelt thanks to everyone who helped increase my understanding of how this is done. I make no claim at all that I could have done it without all that help.

This guide is expressly placed in the public domain, but with the firm expectation that any copying or usage will be credited. (Thanks.)

0.7 City of Heroes vs. City of Villains

As far as I know, both games are identical in their use of binds and macros, and almost all commands are interchangeable between the games. There are a very few commands that are peculiar to one or the other (mostly, to CoV alone) and a number of commands that have different names but appear to be synonyms – you can “lackey” someone in CoH if you wish, and “sidekick” them in CoV.

0.8 Things To Come – Future Plans

This guide will likely never be “complete” since there are always hidden or unknown commands or tweaks, and changes with every Issue and running fix by the Developers. Some of the things in the ongoing agenda include:

- Thorough testing of commands I was not able to test – mostly, those to do with groups, supergroups and the global chat feature. Anyone who does a lot of teaming or SGing, and is messing with the new global chat stuff, is invited to test and correct the commands listed here, and pass along nifty things they find.
- Adding more “cool bind” info. Right now, I'm putting the useful and fun binds I find and hear about into a separate web page on HEROICA!, but there's room to add sections on lists of example binds here.
- Adding more detail on groups of slash commands and how to put them to good use. First up will be the UI, window and display commands, to help you improve your visual gaming experience.
- More details, corrections and data for things like variable and window names and hidden commands and command features. Send 'em along!
- A word about complex bind sets: I have intended to bring in much info about pet, healer, defender and controller binds, but each is a topic for its own guide. There are in particular some very good Mastermind pet bind guides out there. Eventually I'll collate, edit and present the material... but not today. In the meantime, check the HEROICA! site for collections of cool bind sets for these archetypes.

SECTION 1 – BASICS

1.1 Overview

I'm going to put all the special terminology in this section – so if you run into an unfamiliar or cryptic term in the later sections, it's either because you didn't read this one or because I slipped and forgot to include it. (Let me know, in that case.) I'm also going to put most of the general, basic information that applies to both keybinds and macros in here, with a few repeats of key items in other sections.

Read this section – it will help you get going much faster and with fewer problems than if you just jump to the how-to sections!

1.2 Terminology

Keybind – A string of game commands “bound” to a single key or mouse button, which will be executed when that key or button is pressed. Also called just a “**bind**.”

Macro – A string of game commands assigned to a Powers tray button, which will be executed when that Power button is clicked or activated via a keypress.

Syntax – The precise rules by which a keybind or macro string is constructed. If a string is constructed wrong – has faulty syntax, that is – it probably won’t work, or at least won’t do what you want it to do.

Toggle – To turn a power on or off, whichever state it isn’t in, with a single command. Most shields and buffs are toggle powers, which you activate with a click of the button and then deactivate with a click of the same button. There are ways to force toggles to the on and off states, no matter what state they are in to begin with.

Window – Any of the individual dialogs, menus, and separate windows that are part of the user interface.

1.3 Entering Keybinds & Macros

Keybinds and macros are entered from within the game, by typing strings into the chat window’s entry line. The current chat channel selected does not matter; you’re going to override the chat function and direct the command to the “console” (the game’s command input window) by typing a foreslash (/) as the first character. You’re not going to forget that slash. Trust me, after the first time you send a bind string out to the entire zone because you forgot the slash, you’re not going to forget the slash.

There are some good rules for entering binds and macros. The first is to park your character in a safe place, so you won’t have to deal with unexpected foes while you’re tinkering. Inside a tram station or store is a good place. Face your character to the wall, a universal multiplayer game announcement that you’re busy with some internal task and don’t want to be interrupted. If you’re going to be at it a while, you might type the command `/hide` into the console to start. This will make you invisible to everyone else in the chat and search windows, so they won’t bother you. (Remember to `/unhide` when you’re done!) Finally, select a safe chat channel, so that if you do screw up, no one will be privy to your bobble. Using the Team channel while you are not in a team is good – if you accidentally send chat message, either nothing will happen or you’ll just get a warning that you’re not on a team.

You can also enter keybinds by editing a text file and then loading it, but that’s an advanced step we’ll cover separately. For now, the easiest way to start entering binds and macros is directly, in the game.

I strongly recommend that you start with a clean, new set of default keybinds (by going to the Controls menu and resetting everything to Default), and then slowly entering your new binds and testing them. You should also save your keybinds to a local text file every time you are about to make a new set of changes, so that you can quickly reload a working set if you mess up something and need to re-Default things.

1.4 Basic Syntax

The basic syntax for a keybind, which is typed into the chat window’s message-entry window, is:

`/bind key command_string`

This will “bind” the specified command string to the specified key. You can bind commands to almost all of the keys on the keyboard, with some limitations. Once a valid command string is successfully bound to a key, any prior assignment to that key is erased and pressing that key will execute the command string.

The basic syntax for a macro, typed in exactly the same way, is:

`/macro macro_name command_string`

This will “bind” the specified command string to a power-tray button with the identifying name specified. Macro names can be one to three letters or numbers, and some punctuation. (Actually, there is no limit to the length of a macro name, but only three characters will fit on a macro button.) Macros can, confusingly, be given identical names, which is not recommended. Once a valid command string is successfully bound to a macro button, activating that button will execute the command string.

The slash at the beginning of those commands is very important: if you don’t include it, you’ll simply send the string out to whatever chat channel you have selected, provoking much humor and wrath from whoever

sees it. (Sending a bind string out into a zone-wide Broadcast is one of the top not-quite-a-newbie tricks. You are allowed to avoid it. See the suggested rules in 1.3.)

1.5 Variables

Binds and macros are a lot more useful if you can insert variables, such as player or foe names, your own name, level and archetype, etc. City of Heroes includes such variables, which may be inserted into any command string in place of fixed text. It is the dollar sign (\$) first character that identifies the label as a variable, which is why you can't use a dollar sign in most macro and bind text strings.

\$archetype	Your player's archetype – Blaster, Tanker, etc.
\$battlecry	The string you've entered in your ID as your battle cry.
\$level	Your player's level – 2, 10, 35, etc.
\$name	Your player's name – Shenanigunner, Wolf Moon, etc.
\$origin	Your player's origin – Natural, Magic, Science, etc.
\$target	The name of your currently selected target, which can be a foe, another player, or an object.

I suspect there are other, new or hidden variables. Updated information will be gratefully received.

1.6 Useful References

There are several useful references for creating binds and macros. Two are included here: Appendix A lists all the currently known slash commands, and Appendix E lists all the currently known emote codes. You'll likely wear out a few copies of both in your gaming time.

More current lists, and many tips and tricks, can be found on CoH-related web sites and in the official CoH forum devoted to binds. Look these resources up for help, ideas, and information I haven't included here.

Perhaps the most useful reference you can have is a copy of the complete default keybinds, which I haven't included here because it's bulky, but easy to get. And that's one of the key (heh, heh) secrets here: Very few keys in CoH are "hard coded" and unchangeable. Nearly all keyboard and mouse commands are "bound" in a changeable manner. You could erase or eliminate nearly every game command from the keyboard (not that that would be very useful, but it also means you can completely, totally rearrange and remap how the commands are used.) Out of the box, the game simply has a default set of binds that move your character, open and close windows, activate powers, etc. Looking at this default list can be very informative.

To get your very own copy of the default key binds, perform the following steps:

- In the game, go to **Menu | Options | Controls** and select "**Reset to Defaults.**" This is recommended if you've done any inexpert tinkering with binds; otherwise, skip this step. If you do, the file you generate will include any changes you've made.
- In the chat window, type:

```
/bind_save_file c:\defaultbinds.txt
```

You can substitute any path and filename you like. Open the file and you'll find a complete list of the default binds and command strings. (When you get more experience, you can edit this file directly, making as many changes as you like and then load it into the game to make all the bind changes at once.)

It will be assumed that you have this file, printed out for reference, as a companion to this guide.

1.7 Editing Keybind Files

Once you start messing with binds, you'll probably want to move on to making wholesale edits rather than laboriously typing in strings in the game. It's pretty simple; you can even do it while you're in the game, subject to some cautionary notes.

First, save your current keybinds as just described above. In the chat window, type:

```
/bind_save_file c:\defaultbinds.txt
```

It's probably best to use the name of the character, so that each file you save and edit is distinct from the others.

Now switch to the Windows desktop and open this file in your favorite editor. Wait, before you do that, save a backup copy of the file, so you can load your "last good state" if you screw up the file.

Edit away. When you're ready to try the commands, switch back to the game and in the chat window, type:

```
/bind_load_file c:\defaultbinds.txt
```

Test away.

Two notes: You should park your character in a very safe place, like the inside of a store or tram terminal, before switching away to the editor. You don't want to come back and wake up dead. You might also want to `/hide` while you're working.

Also, you may find that switching in and out of the game messes up your mouse control. In this case, go to Control Panel, open your mouse applet, and be sure that "Disable acceleration in games" is unchecked. If it's unchecked, check it. One of those should keep the annoying problem of your mouse going to one-tenth control speed from happening.

SECTION 2 – KEYBINDS

2.1 Keybind Overview

To recap things you should have read above:

- A keybind binds one or more slash commands to a single key. When that key is pressed, the command string will be executed.
- You enter keybinds by typing them into the chat entry window, prefaced by a foreslash (/), in the form:

```
/bind key command_string
```

- The command string should normally be enclosed in one set of double quotes, although they can be omitted for single-word commands.
- Any binds you enter will erase and overwrite any existing bind on that key.

And while we're at it:

- You can erase a keybind, either one you've entered or a default one, by using the "nop" (no operation) keyword:

```
/bind key nop
```

2.2 Key Names

Nearly every key on the keyboard can be used for binds, but, like magical spells, you have to know each key's "true name" – which might not be obvious. For example, to bind something to the equals key, you can't use "=" – it won't work. You have to use "equals" instead. Many keys have similarly odd, but sensible once you understand them, names.

The list of allowable key names can be found in Appendix D. This list may be incomplete; more information is actively solicited!

The alphabetic keys are case-insensitive; binding to R and r is exactly the same.

2.3 Basic Command Usage & Command Modifiers

In some cases, all that needs to be done to use a slash command in a keybind is to type the name of the command:

```
/bind F "follow"
```

Note that the command string is in quotes; although you can sometimes get away without the quotes, you should make it a practice to always use them, even when the command is a single keyword, as here. This command, which mimics the default bind for the F key, will cause your character to follow the selected target. However, the following example:

```
/bind A "left"
```

won't do quite what you think (what the default bind for the A key does). Since hardware and operating system key repeats are disabled within City of Heroes (actually, they are discarded everywhere except in the chat text entry window), pressing A with this bind will cause your character to move the default amount in a strafe-left manner. And stop. Since what you probably want is for the character to keep strafing left as long as you hold the key, you have to add a modifier:

```
/bind A "+left"
```

It's that + that makes the key repeat the action as long as it's held down.

Now suppose you want to toggle on a power or state – like autorun (R in the default key mapping). If you use

```
/bind R "autorun"
```

what you'll get is a status response: you'll see "autorun 0" in the chat window, since the above command is treated as an inquiry into the state of the autorun command. If you try:

```
/bind R "+autorun"
```

you'll get autorun as long as the key is held down... or the same as holding down the W key, not very useful. To make autorun toggle on and off the way the default is mapped, you have to use:

```
/bind R "++autorun"
```

...and there's the trick. The ++ tells the game that it's a toggle command: each press will toggle the state of that power on or off. If you were to be silly and use:

```
/bind Q "++turn_left"
```

what you would get is your character spinning in left circles when you pressed Q, until you pressed Q again to stop it. Silly, but again not very useful.

Commands that toggle can usually also accept a numeric toggle code. For instance:

```
/bind R "autorun 1"
```

would force autorun on, no matter how many times it was pressed. You could then bind another key:

```
/bind V "autorun 0"
```

to turn autorun off unambiguously. This isn't a very useful example, since toggling autorun on and off with one key is quite enough for most players, but there are many situations where you want a firm "on" command and a firm "off" command, with no possibility of, say, dropping your shields during a battle, or turning off Hover or Fly in a sticky situation.

We'll go into more detail about toggles later.

Note also in all these examples that there is no slash in the bind string, except at the beginning. A slash is put in front of a slash command only when it is being executed directly, by itself, from the console line (which is rare except for user interface commands). It's that first slash that tells the game that what follows is a console command of some sort.

2.4 Command Separators

The real power of binds and macros isn't in binding a single command to a key or macro button: it's in the ability to string multiple commands together in that bind. There are some limitations in how you can combine actions – mainly, you cannot easily combine two attack powers into one bind or macro – but generally any reasonable combination of actions can be made. If there is a limit to the length of a bind command string, it's long enough that it will rarely be a problem.

Here is perhaps the single most useful custom bind for melee types:

```
/bind G "target_enemy_near$$follow"
```

This extremely useful bind causes your character to target the nearest foe and follow (lock onto) them. By binding it to my G key, I have the option of tapping F to follow a selected foe (useful when I want to home in on a boss surrounded by minions who might be closer to me), or G to just pounce on the closest foe. In the middle of a fight, surrounded by foes, it is a huge timesaver (and occasionally a butt-saver) to be able to whack G and retarget the nearest foe, rather than the one the game selected for me (who might be out of point-blank range).

The trick here is the "\$\$" characters, which act as a separator between commands. If you were to simply type a list of commands separated by spaces, the console would be unable to parse the line and while it might do something, it's not likely to be what you wanted. So each command needs to be separated from the next by a "\$\$" pair, with **no** spaces around it.

You can string multiple commands together using the \$\$ separator, but there are often limitations on which commands will work in certain cases and sequences. You will probably have to experiment with each new combination to find one that works the way you want it to. Here's a simple mod to the above bind that can be helpful in a team situation:

```
/bind G "target_enemy_near$$g I've got the $target!$$follow"
```

This bind will target the nearest enemy, announce in the Team channel "I've got the Bone Daddy!" (or whichever foe was targeted, by name), and then follow him. Since the chat text is only in the Team channel and simply won't show up when you're not teamed, it won't bother non-team players.

And... **ahem**. A word about that. It's an annoying newbie trick to put a chat message on your power activations; no one you're not teamed with cares that you've activated Fly, hurled a Zapp, turned on your Plasma Shield, etc. Most newbies who discover the joys of chat-binding do it.. once. And get howled out of the zone, most likely. Don't be a clueless jerk; don't bind chat messages to your powers except very selectively in the Team channel, when the message will be helpful – every single time! – to your mates.

A useful variant of this example bind, although it's hard to make it fully automatic, is:

```
/bind H "g I'm assisting $target!$$follow"
```

If you click on a teammate you wish to assist (for example, a tanker pounding on a boss) and then hit H, you will announce to your teammates, "I'm assisting Shenanigunner!" and follow that mate as he moves from target to target. There are some limitations on this bind, but it might be helpful to some players. The `/assist` command might be useful in a bind like this, too.

2.5 Toggles and Forced Toggles

One of the problems with keybinds is that most are, by default, a toggle – the bind will simply turn the power to whichever state it's not in. Sometimes, as with the autorun key, that's exactly what you want. Other times, you want an absolute, guaranteed "power on" or "power off," even if you hit the key by mistake.

Easy enough. There are several "power activation" commands that operate in different ways, and it's easy to select the one you want.

You can toggle a power by specifying its name (preferred) or which tray slot it resides in:

```
/bind P "powexec_slot 3"  
/bind P "powexec_name Fire Shield"
```

Assuming Fire Shield was in slot 3 of the main tray, these binds would work exactly the same – pressing P would toggle Fire Shield on and off. (I can think of some uses for the slot-number method, but in general, you should stay with the power-name method.)

But if you want Fire Shield to go on, and on only, when you whack a specific key, so that you never inadvertently drop the shield during a battle, you would use:

```
/bind P "powexec_toggleon Fire Shield"
```

Which would always force Fire Shield on, even if it was already on. (That is, if the power is on, the command would have no effect.) You could turn the power off by clicking its tray button, or by adding a forced off bind:

```
/bind O "powexec_toggleoff Fire Shield"
```

SECTION 3: MACROS

3.1 Macro Overview

If you've read this far, macros are simple: they are exactly like keybinds in every way, except that they are bound to a Power tray button instead of a keyboard key. The only difference is that the basic syntax is:

```
/macro AST "g I'm assisting $target!$$follow"
```

which will create a button labeled AST in the first open power tray slot. Clicking this button, or activating it with an associated keypress, will be exactly the same as pressing H in the above keybind example.

Macros are the primary reason you have 90 power tray slots. Besides being able to create a couple of alternate power configurations, you can create any number of macro trays – one for soloing, one for team work, one to primarily control or defend, one for melee or ranged attack work, etc.

The only other useful thing to say about macros, except for what's already been said under the previous section, is that there is a second macro creation command, `/macroslot`. This command assigns the newly created macro to a specific slot instead of the first available one:

```
/macroslot 5 AST "g I'm assisting $target!$$follow"
```

...would put the new macro in slot 6 of the primary tray. (Yes, the numbering for this command is zero-based, so 0-9 correspond to slots 1-0.) I'm not sure this has any great usefulness, as you can create a macro and then drag it anywhere you like. But there you go.

Macros may be named with any combination of letters and numbers from one character to... many, I'm not sure what the limit is. However, more than three characters will not fit on the button, so you would be wise to keep your macro names to three characters or less.

Appendix A: Slash Commands Reference

Keywords in *italics* represent values to be specified.

Elements in [brackets] are optional. If an element is not in brackets, it is required.

Numbers in {braces} are required:

- Numbers separated by vertical bars {0|1} represent the valid selections.
- Numbers separated by a dash {1-4} or {0.1-2.0} indicate the range of acceptable values.
- Some commands that require a numeric value will return the current state if entered without a number; others will return an error message.

Commands that use an underscore (*_*) to separate words can also be entered without the underscore, for example, */window_hide* and */windowhide* are equivalent. The underscore versions are generally used here. All commands are also case-insensitive; UPPERCASE and CamelCase words are only for convenience.

I have listed all known synonyms, but entered the usage information only for the one I think is the most useful or obvious option, and pointed the other terms to it.

Current Status: Most of the global chat stuff is untested. Verifications and corrections solicited! Anything **highlighted in blue** is something I have not yet verified or which I have found to be buggy – so use it cautiously and be sure to tell me anything useful you find out about it.

<i>/ac string</i>	See arena
<i>/afk message_string</i>	Marks the player as Away From Keyboard. If no string is specified, a little balloon with “AFK” in it appears over your character’s head. Otherwise, the string is displayed there. Note that an auto power like Hasten can interrupt an AFK status. Note also that this command is how to put a text bubble up while you’re typing a chat message... see the Heroica! site for details.
<i>/ai string</i>	See arenainvite
<i>/alttray {1-9}</i>	Activate the power in the specified slot of the current secondary tray.
<i>/alt2tray {1-9}</i>	Activate the power in the specified slot of the current tertiary tray.
<i>/alttraysticky</i>	Toggle the secondary trays in and out of visibility. This command cycles between the base tray, base+secondary tray, base+secondary+tertiary tray, and base tray again.
<i>/arena message_string</i>	Send message on the Arena chat channel. (Synonym <i>ac</i>)
<i>/arenainvite name</i>	Invite player to join arena event. (Synonym: <i>ai</i>)
<i>/assist</i>	Set your current target to the selected ally’s target.
<i>/assist_name name</i>	Set your current target to the named ally’s target.
<i>/auction string</i>	See request
<i>/autoperf {0 1}</i>	Automatically change world detail for performance. <i>Function unclear.</i>
<i>/autoreply</i>	Start a chat reply including the character of the bind key used to summon it. Works only for keybinds and not as a slash command.
<i>/autorun {0 1}</i>	Toggle autorun on and off. Usually bound to the R key.
<i>/b message_string</i>	See broadcast
<i>/backward</i>	Move backwards. Usually bound to the S key.

<code>/beginchat message_string</code>	Starts chat-entry mode with given string. See also <code>startchat</code> .
<code>/bind key commandstring</code>	Binds a key to a command string. See the rest of this guide for details.
<code>/bind_load</code>	Reads a list of keybinds from <code>keybinds.txt</code> in the default CoH directory.
<code>/bind_load_file filespec</code>	Reads a list of keybinds from a specified file location and name. As of Issue 12 or so, this command started echoing the file load to the status window.
<code>/bind_load_file_silent filespec</code>	Reads a list of keybinds from a specified file location and name. Functions like the old command, without an echo to the screen. Probably best for loading rolling bind sets.
<code>/bind_save</code>	Saves all keybinds to <code>keybinds.txt</code> in the default CoH directory.
<code>/bind_save_file filespec</code>	Saves all keybinds to specified file location and name. As of Issue 12 or so, this command started echoing the file load to the status window.
<code>/bind_save_file_silent filespec</code>	Saves all keybinds to specified file location and name. Functions like the old command, without an echo to the screen.
<code>/bloomscale {2 4}</code>	Sets bloom blur size. Valid values 2 or 4 only.
<code>/bloomweight n</code>	Sets bloom scale. Valid values 0.0 – 2.0.
<code>/broadcast message_string</code>	Send message to entire zone. (Synonyms <code>by yell</code>)
<code>/buffs {0 1}</code>	Toggle display of member buffs in the team list.
<code>/bug subject_string</code>	Report a bug to the developers. Specify a concise subject; you will be given a window to enter additional text.
<code>/buy_coh</code>	Opens the buy City of Heroes window. Note that this announces you are playing on a trial copy of the game even when your copy is registered.
<code>/c message_string</code>	See <code>coalition</code>
<code>/camdist {0-120}</code>	Sets the distance in feet that the third person camera pulls back behind the player. 0 equals first-person view; the upper limit was recently changed from very large (in I5 and previous) and 65 (in I6) to about 120 in I7.
<code>/camdistadjust</code>	Adjusts the camera distance relative to the current camera distance. Reads mousewheel for input. <i>Probably not useful in console commands, as it appears to be permanently bound to the mouse wheel.</i>
<code>/camreset</code>	Resets the camera to a few feet behind the player, looking forward. Bound to the PAGEDOWN key by default.
<code>/camrotate</code>	Camrotate (bound to PAGEUP by default) allows controlled camera rotation around the player. The bound key must be pressed while the view is rotated with the mouse. This command should be bound to a suitable key, and not invoked through the console.
<code>/camturn</code>	Turns the camera to match player facing direction. Similar to <code>camreset</code> except that camera distance is not reset. See also <code>playerturn</code> .
<code>/canlook {0 1}</code>	Toggles “mouselook,” which permits the character to look around using the mouse instead of moving the in-game pointer.
<code>/cc {0-4}</code>	See <code>costume_change</code>

<code>/cce emotestring {0-4}</code>	See <code>cc_emote</code>
<code>/cc_emote emotestring {0-4}</code>	Combines an emote and a costume change. (Synonym: <code>cce</code> .) (Not yet functional in I14.)
<code>/chan_create channel</code>	Create a new chat channel.
<code>/chan_desc channel string</code>	Set chat channel's description to <i>string</i> .
<code>/chan_invite channel</code>	Invite player or chat handle to a chat channel.
<code>/chan_invite_gf channel</code>	Invite your entire global friends list to a chat channel.
<code>/chan_invite_sg channel rank</code>	Invite your entire supergroup to a chat channel. Only leaders may use this command. You can invite members by rank: <ul style="list-style-type: none"> 0 – Invite all supergroup members. 1 – Invite captains and leaders only. 2 – Invite leaders only. See also <code>ginvite_sg</code>
<code>/chan_invite_team channel</code>	Invite your entire team to a chat channel.
<code>/chan_join channel</code>	Join an existing chat channel.
<code>/chan_leave channel</code>	Leave a chat channel.
<code>/chan_members channel</code>	List all members of channel.
<code>/chan_mode channel options</code>	Changes default access rights for new user who joins the channel. Valid Options: <ul style="list-style-type: none"> -join kicks user from channel +send/-send gives/removes user ability to send messages to channel +operator/-operator gives/removes operator status from another user in the channel
<code>/chan_motd channel string</code>	Set the channel's Message Of The Day, which is sent to everyone that joins the channel.
<code>/chan_send channel string</code>	Send message to chat channel. You must be in the channel and have Send privileges. (Synonym: <code>send</code>)
<code>/chan_user_mode channel name options</code>	Sets user permissions for specified user on channel. You must have operator status to set permissions. Valid Options: <ul style="list-style-type: none"> -join kicks user from channel +send/-send gives/removes user ability to send messages to channel +operator/-operator gives/removes operator status from another user in the channel
<code>/change_handle name</code>	Change your global user name, if allowed. There are limits on how often a global handle can be changed (e.g., it may be a one-time change for some users), so use this with caution.
<code>/chat</code>	Toggles the chat window. (Synonyms: <code>toggle chat</code> , <code>window_toggle chat</code>)
<code>/chat_beta {0 1}</code>	Permit participation in the Chat Server Beta Testing. (Obsolete.)
<code>/chat_cycle</code>	Cycles through the default chat channels.

See Appendix G for complete information on using chat save and load functions.

<code>/chat_load</code>	Reads a saved chat configuration (tabs, channels, names) from the CHAT.TXT file in the default installation folder.
<code>/chat_load_file filename</code>	Reads a saved chat configuration (tabs, channels, names) from the specified file name in the default installation folder, or, from the file on another path if it is specified.
<code>/chat_save</code>	Saves the current chat configuration (tabs, channels, names) to the CHAT.TXT file in the default installation folder.
<code>/chat_save_file filename</code>	Saves the current chat configuration (tabs, channels, names) to the specified file name in the default installation folder, or, to the file on another path if it is specified.
<code>/chat_set channel</code>	Sets the channel to the given string. Works only for global channels, not defaults.
<code>/chatoptions {0-4}</code>	Toggles chat options for specified window.
<code>/ci</code>	See <code>coalition_invite</code>
<code>/clearAttributeView</code>	Clear the attribute target. (Command accepted but function unclear.)
<code>/clearchat</code>	Clears all chat buffers – equivalent to executing “Clear History” in each chat tab.
<code>/clear_petnames</code>	Clears all names of all your named pets.
<code>/clearRewardChoice</code>	Choose “no reward” in the current reward choice list.
<code>/clicktomove {0 1}</code>	Enable and disable click-to-move. Requires argument, unlike <code>ctmtoggle</code> . When enabled, clicking on any non-clickable point will create a pretty crystalline cursor, and your character will move to it. Maximum move range is about 60 yards. Useful for zooming around missions and such. (Synonym: <code>ctm</code>) See also: <code>ctminvert</code> , <code>ctmtoggle</code>
<code>/clear_tray</code>	Clears all power trays, except for macros.
<code>/cmdlist</code>	Displays all console commands available in the system chat window. (Useful for finding updates and changes to this list – turn on chat logging first to save to a text file.)
<code>/coalition message_string</code>	Send message to the coalition chat channel. You must be a member of a supergroup that is in a coalition with another group for this function to work. (Synonym: <code>c</code>)
<code>/coalition_cancel</code>	Cancel coalition with selected supergroup.
<code>/coalition_invite player_name</code>	Invites the named player to join a coalition. The player must be the leader of a supergroup for the function to work. (Synonym: <code>ci</code>)
<code>/coalition_mintalkrank</code>	Set the minimum rank of a supergroup who your supergroup can hear. (Values?)
<code>/coalition_nosend</code>	Stop your supergroup from sending coalition chat to an ally supergroup.
<code>/coalition_sg_mintalkrank</code>	Set the minimum rank of a your supergroup who can talk on the coalition chat. (Values?)
<code>/comment string</code>	Sets your group-search string to <i>string</i> . (Not functional in I14 – command accepted but has no effect.)

<code>/compatiblecursors {0 1}</code>	Shows the status of selection of standard Windows cursors instead of graphical cursors. (The Windows cursors are not as flexible and don't change color but may work better on some systems.) This command cannot be used to set the option, which must be set on the command line at game startup.
<code>/contextmenu menu_num</code>	Activate a context menu slot. (Command accepted but has no effect; function unclear.)
<code>/copychat tab_name</code>	Copy the entire chat history from specified chat tab into the clipboard. Useful for saving extended game info passed on by other players, or abuse.
<code>/copydebuginfo</code>	Gathers debug info, prints it and copies it into the clipboard. (Command accepted but does nothing. May be obsolete or for use at customer service direction only.)
<code>/costume_change {0-3}</code>	Change costume. Note that this is another of the anomalous zero-based commands! (Synonym: <code>cc</code>) See also <code>cc_emote</code> for a more elaborate option.
<code>/ctm {0 1}</code>	See <code>clicktomove</code>
<code>/ctm_invert {1 0}</code>	Functionally identical to <code>clicktomove/ctm</code> except that the enable value is reversed (0=enabled). Argument required.
<code>/ctm_toggle</code>	Toggles click-to-move status. No argument allowed. Bug: Displayed enable/disable message can get inverted if used with other CTM change commands. Also seems to hard-override other CTM settings.
<code>/cursorcache {0 1}</code>	Enable cursor cache for smoother cursor changes.
<code>/demorecord filename</code>	Begin recording a demo.
<code>/demostop</code>	Stop demo record/play.
<code>/demote name</code>	Demote supergroup member one rank.
<code>/dialog_answer string</code>	Answer dialog with button matching provided text. (Command accepted but function unclear.)
<code>/dialog_no</code>	Answer OK, No, or Cancel to current dialog.
<code>/dialog_yes</code>	Answer OK, Yes, or Accept to current dialog.
<code>/disable2D {0 1}</code>	Disables 2D sprite drawing. (Main, and perhaps only effect seems to be to turn all UI elements on and off.)
<code>/dofweight</code>	Sets DOF scale. Valid values 0.0 – 2.0. See also <code>usedof</code> .
<code>/down</code>	Move down (if flying). Bound to the X key by default.
<code>/e emotestring</code>	See <code>emote</code>
<code>/e3screenshot {0 - ?}</code>	Enables special e3 2004 screenshot mode. (Values and function unknown.)
<code>/em emotestring</code>	See <code>emote</code>
<code>/emaildelete message_number</code>	Delete message [message number].
<code>/emailheaders</code>	Request email headers. (Command accepted but has no effect; function unclear.)
<code>/emailread message_number</code>	Request message [message number].
<code>/emailsend name subject message_string</code>	Send message [player names] [subject] [body].

<code>/emote <i>emotestring</i></code>	Causes player to display an emote animation or emote string. Emote codes can be found in Appendix E. Any string that does not match a valid emote code will be displayed in a visible thought bubble for a few seconds. (Synonyms: e em me)
<code>/estranger <i>name</i></code>	See unfriend
<code>/ex <i>name</i></code>	See sidekick
<code>/exemplar <i>name</i></code>	See sidekick
<code>/f <i>message_string</i></code>	Talk to friends channel.
<code>/face</code>	Turn player to face selected target.
<code>/findmember</code>	Search for player. Appears identical to <code>search</code> .
<code>/first {0 1}</code>	Toggles between first and third person camera. (Inverse of <code>third</code> .)
<code>/fl</code>	See <code>friendlist</code>
<code>/follow</code>	Toggle follow mode.
<code>/forward</code>	Move forward.
<code>/forward_mouse</code>	Move forward; enable autorun after 2 seconds.
<code>/friend <i>name</i></code>	Add player to friend list.
<code>/friendlist</code>	Display friend list in chat window. (Synonym: <code>f1</code>)
<code>/fsaa {0 2 4}</code>	Sets the amount of full screen antialiasing. Other values can be set but their impact is unclear. <i>Note: FSAA has more impact on framerate than nearly any other graphics setting!</i>
<code>/fullscreen {0 1}</code>	Effect not entirely clear. Sets video mode to fullscreen. If set to 0, the game will start in windowed mode next time; when set to 1, game will start in fullscreen mode. Cannot be changed during gameplay; you have to make this setting and then restart to change the view. See also <code>maximize</code> .
<code>/g <i>message_string</i></code>	See <code>group</code>
<code>/gamereturn</code>	Reset UI by leaving in fullscreen mode, closing dialogs and closing all secondary (nonessential) windows. (Same as <code>windowcloseextra</code>)
<code>/getarenastats</code>	Get your arena combat statistics.
<code>/getallarenastats</code>	Get your arena combat statistics, more comprehensive display.
<code>/getratedarenastats</code>	Get your arena combat statistics for rated matches.
<code>/getcomment</code>	Get your group-search string. (Command accepted but returns blank string in all cases.)
<code>/getglobalname <i>charname</i></code>	Get player's global name from character name. Using command without name returns an "unknown command" error.)
<code>/getglobalsilent <i>charname</i></code>	Get player's global name from character name without reporting results to chat window. Appears to work but if no display is generated, not sure of purpose...
<code>/getlocalinvite <i>globalname</i></code>	Invite current character name from global player name.
<code>/getlocalname <i>globalname</i></code>	Get currently active character name from global player name.
<code>/getpos</code>	See <code>loc</code>
<code>/gfriend <i>name</i></code>	Add a player to your global friends list.

<code>/gfriends</code>	Display all members of your global friends list.
<code>/gfriend <i>player name</i></code>	Add player to global friends list via their player name. (Not functional?)
<code>/ghide</code>	Make yourself invisible to your global friends. Disconnected from <code>hide/unhide</code> .
<code>/gignore <i>name</i></code>	Ignore user on global chat.
<code>/gignoring</code>	Display list of ignored users on global chat.
<code>/ginvite <i>player_name</i></code>	Invites the named player to join a global chat.
<code>/ginvite_sg <i>channel rank</i></code>	Invite your entire supergroup to a global chat channel. Only leaders may use this command. You can invite members by rank: <ul style="list-style-type: none"> 0 – Invite all supergroup members. 1 – Invite captains and leaders only. 2 – Invite leaders only. See also <code>chan_invite_sg</code>
<code>/gmotd</code>	Recall the global message of the day, as displayed at first login.
<code>/goto_tray {1-9}</code>	Set the main tray to the specified tray number.
<code>/goto_tray_alt {1-9}</code>	Set the secondary tray to the specified tray number.
<code>/goto_tray_alt2 {1-9}</code>	Set the tertiary tray to the specified tray number.
<code>/goto_trays_tray {1-3} {1-9}</code>	Set the specified tray (1-3) to the desired tray number (0-9).
<code>/group <i>message_string</i></code>	Send message to group channel. (Synonyms: <code>g</code> , <code>team</code>)
<code>/guide</code>	See <code>helpchat</code>
<code>/gunfriend <i>name</i></code>	Remove a player from your global friends list. (Via <code>global name</code> ?)
<code>/gunfriend <i>player name</i></code>	Remove player from global friends list. (Not functional?)
<code>/gunhide</code>	Make yourself visible to your global friends. Has been disconnected from <code>hide/unhide</code> .
<code>/gunignore <i>name</i></code>	Un-ignore user on global chat.
<code>/h</code>	See <code>helpchat</code>
<code>/hc</code>	See <code>helpchat</code>
<code>/help</code>	Open Help window.
<code>/helpchat</code>	Selects the global Help channel. (Synonyms: <code>guide</code> , <code>h</code> , <code>hc</code>)
<code>/hide</code>	Hide your name from other users in the “who’s on” lists. Does not affect <code>ghide/gunhide</code> .
<code>/hideall</code>	Hide your name from other users in all of the “who’s on” lists.
<code>/hidefriends</code>	Hide your name from other users in the Friends list.
<code>/hidegchannels</code>	Hide your name from other users in the Global Channels list.
<code>/hidegfriends</code>	Hide your name from other users in the Global Friends list.
<code>/hideinvite</code>	Hide your name from other users in invites.
<code>/hidesearch</code>	Hide your name from other users in search.
<code>/hidesg</code>	Hide your name from other users in the Supergroup list.
<code>/hidetell</code>	Hide your name from other users in tells (direct chat).
<code>/hideprimarychat</code>	Toggle primary chat window text messages. You can reduce the chat window to just the chat entry line with this command.

<code>/i name</code>	See invite
<code>/ignore name</code>	Ignore user.
<code>/ignore spammer name</code>	Ignore user as spammer (automatically reports name as spammer).
<code>/ignorelist</code>	Displays a list of ignored users.
<code>/info</code>	Displays the information on a selected item, same as right-clicking and selecting Info from the pop-up menu.
<code>/info_self</code>	Displays your own information, the same as others see when they "info" you.
<code>/info_self_tab {0-5}</code>	<p>Displays your own information, the same as others see when they "info" you, opening the window to the named info tab. Tabs are referenced by number:</p> <ul style="list-style-type: none"> 0, 1 – Description. 2 – Powers. 3 – Badges. 4 – Arena. 5 – PvP. <p>Note that the ordering of the PvP and Arena tabs are reversed – this may be a minor bug to be fixed in later updates.</p>
<code>/info_tab {0-5}</code>	<p>Displays the information on a selected item, same as right-clicking and selecting Info from the pop-up menu, opening the window to the named info tab. Tabs are referenced by number:</p> <ul style="list-style-type: none"> 0, 1 – Description. 2 – Powers. 3 – Badges. 4 – Arena. 5 – PvP. <p>Note that the ordering of the PvP and Arena tabs are reversed – this may be a minor bug to be fixed in later updates. Display of the Powers tab is suppressed in PvP.</p>
<code>/insp_combine inspname1 inspname2</code>	Combines three of the first named Inspirations into one of the second name. You must put quotes around multi-word Inspiration names, e.g. "break free" or "catch a breath".
<code>/insp_delete inspname</code>	Delete named Inspiration.
<code>/inspexec_name inspname</code>	Activate an Inspiration by name.
<code>/inspexec_pet_name inspname petname</code>	Activate a named Inspiration on a pet by pet name.
<code>/inspexec_pet_target inspname</code>	Activate a named Inspiration on the targeted pet.
<code>/inspexec_slot column</code>	See inspirationslot
<code>/inspexec_tray row column</code>	Activate an inspiration slot in the specified row and column.
<code>/inspirationslot column</code>	Activate an inspiration slot in the first row of the specified column.
<code>/invite name</code>	Invite player to join team. (Synonym: i)
<code>/k name</code>	See kick
<code>/keybind_reset</code>	Resets all keybinds to default. See also unbind, unbind_all. Use with caution!

<code>/kick name</code>	Kick player from team. (Synonym: k)
<code>/kiosk string n</code>	If you're within range of an Info kiosk, this will pop up the "home" info page. The string and n values are required but appear to have no effect. Overall it appears to be a broken or incomplete command. The kiosks no longer appear to be working in current Issues.
<code>/l message_string</code>	See local
<code>/lackey</code>	Invite selected player to be your lackey. (Synonym: lk)
<code>/leaveteam</code>	Quit your current team.
<code>/left</code>	Strafe left. Bound to A key by default.
<code>/levelingpact playername</code>	Invite named player to join a Leveling Pact.
<code>/lfg [0 1]</code>	Toggle LFG (looking for group) status. (Command accepted but appears to have no effect.)
<code>/lfgset {0 1}</code>	Set LFG (looking for group) status. (Command accepted but appears to have no effect.)
<code>/lightmaplodscale {0.0 ?}</code>	Set lightmap LOD scale. (Obsolete.)
<code>/link_channel channelname</code>	Activates context menu for named channel.
<code>/link_info</code>	Provides info window for named channel. (Command accepted but appears to have no effect.)
<code>/link_interact playername</code>	Activates context menu for named player interactions.
<code>/link_interact_global arg arg</code>	Activates context menu for global player name. (Command generates error message but arguments unknown.)
<code>/lk</code>	See lackey
<code>/loc</code>	Get current position. (Synonym: getpos)
<code>/local message_string</code>	Send message to anyone in your immediate area, about a 250 foot radius. (Synonym: l)
<code>/localtime</code>	Displays (your computer's) local time.
<code>/lodbias {0.0 2.0}</code>	Multiplier for LOD (Loss of Detail) distances for entities. The default is 1.0. Setting this to 0.5 will cause detail switches to happen at half the distance; 2.0 will cause switches to happen at twice the default distance. Lower values improve performance; higher ones increase your character's vision. Appears to be obsolete; see DOFWeight.
<code>/logchat</code>	Toggle chat logging. Chat logs appear by date in the \logs folder under the main CoH folder.
<code>/lookdown {0 1}</code>	Moves look angle down. Normally, this command and lookup are used with the + and ++ modifiers to permit controlled up and down looking. Works in conjunction with lookup to control free look capability. (If both lookdown and lookup are set to 1, or both are set to 0, you will have free look capability. Setting one or the other to 1 will force the view to a straight up or straight down view, persistent against changes. There must be some use for these settings, but I can't figure it out. I think it's a slightly buggy side effect.)
<code>/lookup</code>	See lookdown
<code>/lp messagestring</code>	Sends message to Leveling Pact channel.
<code>/macro macroname command_string</code>	Add a macro to first empty slot. See the rest of this guide for details.

<code>/macroslot slotnum macroname command_string</code>	Add a macro to the specified slot of the primary tray. Bug warning: the slot numbering for this command is zero-based, so the first slot is 0, second is 1, last is 9. All other slot commands are one-based!
<code>/makeleader name</code>	Designated new team leader. Can be used only by current leader. (Synonym: ml)
<code>/mal</code>	See malefactor
<code>/malefactor</code>	Invite selected player to be your malefactor. (Synonym: mal)
<code>/manage</code>	Open the Enhancement management window. (This appears to be the only menu/window name that does not work in the other window-control commands.)
<code>/map</code>	Toggles the map window. (Synonym: toggle map, window_toggle map)
<code>/maxfps {1-?}</code>	Set the maximum FPS (frames per second) rate. This used to be capped at 30 but now appears to accept any value. Normally you will want this maximized (at 30 or higher; 0 appears to be "no limit"), but it may be useful in some circumstances to enter a slower rate. Very slow rates (under 5) are NOT recommended!
<code>/maximize {0, 1}</code>	Effect is unclear. Compare with fullscreen.
<code>/maxinactivefps {1-30}</code>	Set the maximum FPS (frames per second) rate while the game is not in the foreground. Reducing this value will lessen the impact on other programs brought forward during gameplay. The rate should be high enough for you to be able to keep track of what is happening – 5-8 fps is recommended.
<code>/me emote_string</code>	See emote
<code>/menu</code>	Opens the main menu. (Synonyms: toggle menu; window_toggle menu)
<code>/missionmake</code>	Activate the My Arcs menu of Mission Search.
<code>/missionsearch</code>	Open the Mission Search window.
<code>/ml name</code>	See makeleader
<code>/mmentry</code>	Choose between making and starting a mission maker story arc.
<code>/monitorattribute</code>	Adds attribute to Attribute Monitor.
<code>/mouse_invert {0 1}</code>	When active, inverts the mouse Y axis (pitch) for mouselook.
<code>/mouse_look num</code>	Command key for mouselook. (Function unclear.)
<code>/mouse_speed {0-6}</code>	Mouse speed scale factor for mouse look. 1.0 is default; values over 3 make control erratic in most cases.
<code>/mousepitchmode {0 1 2}</code>	Set mouse pitch mode: (Obsolete.) 0 – Free look. 1 – Return to center after release. 2 – Always centered. (Confusing, not recommended.)
<code>/myhandle</code>	Display your global chat handle.
<code>/nameCaptain name_string</code>	Renames the 'Captain' supergroup rank.
<code>/nameCommander name_string</code>	Renames the 'Commander' supergroup rank.
<code>/nameEnforcer name_string</code>	Renames the 'Enforcer' supergroup rank.

<code>/nameFlunky name_string</code>	Renames the 'Flunky' supergroup rank.
<code>/nameLeader name_string</code>	Renames the 'Leader' supergroup rank.
<code>/nameLieutenant name_string</code>	Renames the 'Lieutenant' supergroup rank.
<code>/nameMember name_string</code>	Renames the 'Member' supergroup rank.
<code>/nameOverlord name_string</code>	Renames the 'Overlord' supergroup rank.
<code>/nameRingleader name_string</code>	Renames the 'Ringleader' supergroup rank.
<code>/nameTaskmaster name_string</code>	Renames the 'Taskmaster' supergroup rank.
<code>/nav</code>	Toggles the navigation window. (Synonyms: <code>toggle nav;</code> <code>window_toggle nav</code>)
<code>/neterrorcorrection {0 1 2}</code>	Adjusts network error correction limits. Details unknown.
<code>/netgraph {0 1 2}</code>	Displays network connection information. Option 1 is low-profile, Option 2 is higher-profile; not sure of other differences.
<code>/newspaper</code>	(CoV) Open your newspaper to see what capers are available, or (CoH) Open the police radio to see what missions are available.
<code>/next_tray</code>	Go to next primary tray.
<code>/next_tray_alt</code>	Go to next secondary tray.
<code>/next_tray_alt2</code>	Go to next tertiary tray.
<code>/next_trays_tray {1-3}</code>	Go to the specified tray's next tray.
<code>/nojumprepeat {0 1}</code>	Disable jump auto-repeat. This means you'll jump only once, no matter how long the key is held down; another jump will require another keypress.
<code>nop</code>	Not really a command, but a null placeholder used to cancel a bind. If you enter <code>/bind x nop</code> , for example, any bind on X will be deleted. Useful for clearing out default binds you don't want.
<code>/norenderthread {0-?}</code>	Function unknown. See also <code>renderthread</code>.
<code>/noreport {0 1}</code>	Do not default to error reporting window on crash. This may suppress the Windows error reporting screen after a crash; confirmation and other purpose unknown.
<code>/nosunflare {0 1}</code>	Disables sun flare (for performance debugging). Removes and restores flare/glare from sunlight (and moonlight?)

[See Appendix G for complete information on using option set, save and load functions.](#)

<code>/option_list</code>	Lists option names.
<code>/option_load</code>	Reads option configuration from the file options.txt in the default installation folder.
<code>/option_load_file filename</code>	Reads option configuration from the specified filename in the default installation folder, or, if specified, in a different location.
<code>/option_save</code>	Saves window configuration to the file options.txt in the default installation folder.
<code>/option_save_file filename</code>	Saves option configuration to the specified filename in the default installation folder, or, if specified, in a different location.
<code>/option_set</code>	Sets an option.
<code>/option_toggle</code>	Toggles an option.
<code>/p name, messagestring</code>	See <code>tell</code>
<code>/petcom stance</code>	Set selected pet to specified action/stance.

<code>/petcom_all stance</code>	Set all pets to specified action/stance.
<code>/petcom_name petname stance</code>	Set named pet to specified action/stance.
<code>/petcom_pow powname stance</code>	Set the stance for all pets cast by the named power.
	For the above commands, the valid pet stances are:
	aggressive – attack any nearby foe without orders.
	defensive – respond to attack by any foe without orders.
	passive – do nothing without orders.
	And the valid pet actions are:
	attack – attack currently selected target.
	dismiss – dismisses pet gracefully.
	follow – follow me.
	goto – go to the selected spot.
	stay – stay at the selected spot.
<code>/petition subjectstring</code>	Add user petition (stuck, cheated, etc.) to the database. This is more for immediate help from a game master than <code>/bug</code> . Give a GM time to get the petition and help you – it can take a few minutes or more.
<code>/pet_select</code>	Select pet.
<code>/pet_select_name petname</code>	Select named pet.
<code>/petoptions</code>	Displays pet window options menu. Warning: may crash CoH if CoV is not enabled.
<code>/petrename petname</code>	Renames selected pet.
<code>/petrename_name oldname newname</code>	Renames named pet.
<code>/petsay string</code>	Have selected pet say or emote string.
<code>/petsay_all string</code>	Have all pets say or emote string.
<code>/petsay_name petname string</code>	Have named pet say or emote string.
<code>/petsay_pow powname string</code>	Have all pets cast by specified power say or emote string.
<code>/playernote playername</code>	Opens note window for specified global player name.
<code>/playernotelocal playername</code>	Opens note window for specified current player name.
<code>/playerturn</code>	Turns player to match camera angle. Does not change camera distance. See also <code>camturn</code> .
<code>/popmenu menuname</code>	Pops up the named menu at the current mouse location. <i>(Command accepted but appears to have no function.)</i>
<code>/powers</code>	Toggles the power inventory window.
<code>/powexec_abort</code>	Cancels the auto-attack power and the queued power.
<code>/powexec_altslot {1-10}</code>	Executes the given power slot from the secondary tray.
<code>/powexec_alt2slot {1-10}</code>	Executes the given power slot from the tertiary tray.
<code>/powexec_auto power_name</code>	Sets the auto-attack power to given named power (or blank to shut it off, or toggles if it's on already).
<code>/powexec_name power_name</code>	Executes a power with the given name.
<code>/powexec_slot {1-10}</code>	Executes the given power slot from the current tray.
<code>/powexec_toggleoff power_name</code>	Toggles a given power off. If its already off, does nothing.
<code>/powexec_toggleon power_name</code>	Toggles a given power on. If its already on, does nothing.

<code>/powexec_tray slot tray</code>	Executes a power in the given slot and tray.
<code>/powexec_unqueue</code>	Cancels the queued power. Bound to the Z key by default.
<code>/prev_tray</code>	Go to previous primary tray.
<code>/prev_tray_alt</code>	Go to previous secondary tray.
<code>/prev_tray_alt2</code>	Go to previous tertiary tray.
<code>/prev_trays_tray {1-3}</code>	Go to the specified tray's previous tray.
<code>/private name, message_string</code>	See <code>tell</code>
<code>/profiler_record filename</code>	Record client profiler information to specified filename. Purpose of this file and function unclear – appears to be for debugging and tech support use.
<code>/profiler_stop</code>	Stop recording client profiler information.
<code>/promote name</code>	Promote supergroup member one rank.
<code>/quickchat</code>	Pops up the quickchat (emotes + chat bubble) menu.
<code>/quit</code>	Quits game to the desktop. Warning: acts instantly!
<code>/r message_string</code>	See <code>reply</code>
<code>/raid_invite</code>	Invites selected player's supergroup to join an instant raid.
<code>/release</code>	Activate medicom unit for emergency medical transport. (Equivalent to clicking "Go to Hospital" button when defeated.)
<code>/release_pets</code>	Deactivate all current pets. (They fall dead, instead of leaving as with the menu "dismiss" command.)
<code>/reloadgfx</code>	Reload all graphics textures. Useful when something has messed up your screen display. Warning: scrambles display for at least a few seconds – do not use in combat.
<code>/renderscale {0.1-1.0}</code>	Changes the scale at which the world is rendered, relative to your screen size. Permits you to keep your screen size the same as desktop, or sufficiently high for well-rendered UI elements, while lowering the effective resolution for performance. This command affects both X and Y scaling simultaneously; see also <code>renderscalex</code> and <code>renderscaley</code> . Not effective unless <code>userenderscale</code> is set to 1. (Setting this value to 0, or cycling <code>userenderscale</code> from 1 to 0, will reset renderscaling to the default of 1.0.
<code>/renderscalefilter</code>	Changes the method of filtering used in renderscaling. Value range and function unknown.
<code>/renderscalex {0.1-1.0}</code>	Changes only the X scaling of the world rendering. See <code>renderscale</code> .
<code>/renderscaley {0.1-1.0}</code>	Changes only the Y scaling of the world rendering. See <code>renderscale</code> .
<code>/rendersize xsize ysize</code>	Changes the size at which the world is rendered. Sizes are specified in x and y screen values and may be "normal" or odd values. Specifying non-multiple values will result in nonlinear x or y scaling.
<code>/renderthread {0-?}</code>	Function unknown. See also <code>norenderthread</code>.
<code>/reply message_string</code>	Reply to last <i>received</i> private message. This has now been differentiated from replying to the last <i>sent</i> private message, which can be replied to using the <code>tell_last</code> command. (Synonym: <code>r</code>)

<code>/req message_string</code>	See request
<code>/request message_string</code>	Send message to request channel. (Synonyms: auction, req, sell)
<code>/requestexitmission {1 n}</code>	Leave mission map once completed. Equivalent to clicking "Mission Completed" text in Nav window. The "1" is required; "0" does nothing. Other values may have other effects – testing is required. Does not set "auto-exit" if called before end of mission.
<code>/respec</code>	Goes to the Respec screen if you have a "free respec" available. Warning: you should only use this command with your character in a safe place – you can be attacked while in this mode.
<code>/right</code>	Strafe right. Bound to the D key by default.
<code>/rsk name</code>	See sidekick
<code>/s message_string</code>	See say
<code>/say message_string</code>	Sends the given text on the current chat channel. (Synonym: s)
<code>/screen x_dimension y_dimension</code>	Sets X and Y screen dimensions. Should be constrained to standard screen dimensions supported by your video card (640x480, 1024x768, 1280x1024, 1600x1200, etc.)
<code>/screenshot</code>	Save a JPEG (.jpg) format screenshot in the \screenshots directory under the default CoH directory.
<code>/screenshottga</code>	Save a Targa (.tga) format screenshot in the \screenshots directory under the default CoH directory.
<code>/screenshottitle filename</code>	Save a JPEG (.jpg) format screenshot in the \screenshots directory under the default CoH directory, using the specified filename. Bug: creates standard screenshot.
<code>/screenshotui {0 1}</code>	Enables or disables the user interface elements for screenshots. If set to 1, the UI will be visible in screenshots; if set to 0, the UI will not be included in screenshots.
<code>/sea</code>	See search
<code>/search</code>	Displays a searchable list of characters with their name, archetype, level, zone and looking for group status. (Synonym: sea)
<code>/sell message_string</code>	See request
<code>/send channel message_string</code>	Send message to the named chat channel. You must be a member of the channel and have send privileges.
<code>/servertime</code>	Displays the current official (game server) time.
<code>/set_title badgename</code>	Set badge title. (Must be one you have, of course.) (Bug: clears currently selected badge title no matter what string is used.)
<code>/sg message_string</code>	See supergroup
<code>/sgcreate</code>	Start a supergroup.
<code>/sgi name</code>	See sginvite
<code>/sginvite name</code>	Invite player to join supergroup. (Synonym: sgi)
<code>/sgk name</code>	See sgkick
<code>/sgkick name</code>	Kick player from supergroup. (Synonym: sgk)
<code>/sgkickyes name</code>	Kick player from supergroup. (Without confirmation?)
<code>/sgleave</code>	Leave your current supergroup.
<code>/sgmode</code>	Toggle supergroup mode.

<code>/sgmodeset</code>	Sets supergroup mode.
<code>/sgraidinvite</code>	Invite selected player's supergroup to join raid.
<code>/sgraidwindow <i>daybits hour</i></code>	Set your supergroup's raid window. (Values unknown.)
<code>/sgsetcostume</code>	Sets supergroup costume parameters. Obsolete?
<code>/sgsetdemotetimeout</code>	Sets supergroup demote timeout.
<code>/sgsetdescription <i>string</i></code>	Sets supergroup description.
<code>/sgsetmotd <i>message_string</i></code>	Sets supergroup MOTD.
<code>/sgsetmotto <i>message_string</i></code>	Sets supergroup motto.
<code>/sgstats</code>	Display supergroup info in chat window.
<code>/shaderdetail {0 1 2}</code>	Controls shader detail level.
<code>/shadowvol {0 1}</code>	Controls whether or not shadow volumes are drawn. Behaves very oddly when set to 1 on some systems.
<code>/show <i>window_name</i></code>	Forces the given window to be shown. (Synonym: <code>window_show</code>) Has no opposite, although <code>/hide</code> is sometimes incorrectly cited.
<code>/showfps {0 1}</code>	Show current framerate as a small boxed number on top right edge.
<code>/shownewtray</code>	Opens a tear-away Tray window. May be repeated to open multiple trays. As with the + button on the main tray, it will open trays beginning with the last one you had open.
<code>/showpetnames</code>	Lists names of all named pets.
<code>/sidekick <i>name</i></code>	Invite player to be your sidekick. (Synonyms: <code>ex</code> , <code>exemplar</code> , <code>rsk</code> , <code>sk</code>)
<code>/sidekick_accept</code>	Accept an invitation to be a sidekick.
<code>/sidekick_decline</code>	Decline an invitation to be a sidekick.
<code>/sk <i>name</i></code>	See <code>sidekick</code>
<code>/slashchat</code>	Starts chat-entry mode and copies whatever key is pressed into the chat buffer. Used by default with "/" but could be used with other keys – to what purpose is not clear.
<code>/speed_turn {1-359}</code>	Set the number of degrees for each increment of rotate left/right.
<code>/ss {0 1}</code>	Controls whether or not simple shadows are drawn.
<code>/startchat</code>	Starts chat-entry mode.
<code>/stopinactivedisplay</code>	Stops rendering when the game is not the foreground application.
<code>/stopmonitorattribute</code>	Removes attribute from Attribute Monitor.
<code>/stuck</code>	Tries to shift your character to the nearest unstuck position; for use when you get stuck between objects or in map flaws. (If it doesn't work, try sending a <code>/petition</code> and waiting a bit to see if a GameMaster will help you.)
<code>/supergroup <i>message_string</i></code>	Send message to super group channel. (Synonym: <code>sg</code>)
<code>/suppressCloseFx [0 1]</code>	Hides all character effects when the camera is closer than the <code>SuppressCloseFxDist</code> setting. Useful when close camera viewpoint is obscured by powers effects, etc.
<code>/suppressCloseFxDist <i>feet</i></code>	Within this camera distance, character effects will be suppressed. The practical limit is 75, maximum viewpoint camera distance.
<code>/sync</code>	Try to resynchronize character/client with game server. Use when character cannot be moved, becomes invisible to teammates, etc.
<code>/synch</code>	see <code>sync</code> .

<code>/t name, message_string</code>	See <code>tell</code>
<code>/tabglobalnext</code>	Cycle forward through all chat tabs in all windows. Will open the corresponding chat window if necessary.
<code>/tabglobalprev</code>	Cycle backwards through all chat tabs in all windows. Will open the corresponding chat window if necessary.
<code>/tabnext {0-4}</code>	Cycle forward through all chat tabs in indicated chat window (0-4).
<code>/tabprev {0-4}</code>	Cycle backward through all chat tabs in indicated chat window (0-4).
<code>/tabselect tabname</code>	Select the given chat tab. Will open the corresponding chat window if necessary.
<code>/tabtoggle</code>	Make the previously active chat tab the new active tab. Used to flip between two tabs.
<code>/target</code>	Toggles the target window. (Synonyms: <code>toggle target</code> , <code>window_toggle target</code>)

For more information on custom targeting, see ['Gunner's Guide to Advanced Targeting on the HEROICA! web site](#). There's a lot of cool stuff you can do with these commands.

<code>/target_custom_...</code>	Powerful customizable targeting comand. There are four versions, which conclude with the following suffixes: <ul style="list-style-type: none"> <code>...near</code> (closest target) <code>...far</code> (farthest target) <code>...next</code> (next target, in near to far order) <code>...prev</code> (next target, in far to near order) Each of these commands can be directed to a specific class of targetable object by one of these keywords: <ul style="list-style-type: none"> <code>enemy</code> (identical to <code>target_enemy...</code>) <code>friend</code> (identical to <code>target_friend...</code>) <code>defeated</code> target with zero hit points <code>alive</code> enemy, friend or NPC with nonzero hit points <code>mypet</code> any pet spawned by you <code>notmypet</code> any pet not spawned by you <code>base</code> not clear – seems to be for targetable objects other than players and NPCs, but experimentation has shown this to be erratic <code>notbase</code> not clear – see base <code>teammate</code> any teammate <code>notteammate</code> any non-teammate player For example, <code>target_custom_near teammate</code> will target the nearest of your teammates, even among other players.
<code>/target_enemy_far</code>	Targets the farthest visible enemy.
<code>/target_enemy_near</code>	Targets the nearest enemy.
<code>/target_enemy_next</code>	Cycles through visible targetable enemies in near to far order. (Synonym: <code>toggle_enemy</code>)
<code>/target_enemy_prev</code>	Cycles through visible targetable enemies in far to near order. (Synonym: <code>toggle_enemy_prev</code>)

<code>/target_friend_far</code>	Targets the farthest friend. A friend is any friendly player or pet, not just teammates.
<code>/target_friend_near</code>	Targets the nearest friend.
<code>/target_friend_next</code>	Cycles through visible targetable friends in near to far order.
<code>/target_friend_prev</code>	Cycles through visible targetable friends in far to near order.
<code>/target_name <i>string</i></code>	Target any entity whose name begins with <i>string</i> .
<code>/team <i>message_string</i></code>	See <code>group</code>
<code>/team_accept</code>	Accepts an invitation to a team.
<code>/team_decline</code>	Declines an invitation to a team.
<code>/team_kick_internal</code>	Kicks a character without warning from team.
<code>/team_quit_internal</code>	Quits team without warning.
<code>/team_select [1-8]</code>	Select team member (by number in team list).
<code>/team_task <i>int int int</i></code>	Select the team task. Function uncertain .
<code>/tell <i>name, message_string</i></code>	Send a message to only one player. (Synonyms: <code>p</code> , <code>private</code> , <code>t</code> , <code>whisper</code>)
<code>/tell_last <i>message_string</i></code>	Reply to the same person to whom you last <i>sent</i> a private message. This has now been differentiated from replying to the last <i>received</i> private message, which can be replied to using the <code>reply</code> command. (Synonym: <code>t1</code>)
<code>/texaniso {0 1 2 4 8 16}</code>	Sets amount of anisotropic filtering. UI permits only those values shown, but other integer values can be entered. Effect of these interim values uncertain.
<code>/texwordeditor <i>texname</i></code>	Edit the text layout for translatable textures. Exact function unknown ; probably not something for users to mess with .
<code>/third {0 1}</code>	Toggles between first and third person camera. (Inverse of <code>first</code> .)
<code>/t1</code>	See <code>tell_last</code>
<code>/toggle <i>window_name</i></code>	Show a window if hidden, hide a window if shown. (Synonym: <code>window_toggle</code>)
<code>/toggle_enemy</code>	See <code>target_enemy_next</code>
<code>/toggle_enemy_prev</code>	See <code>target_enemy_prev</code>
<code>/trade <i>name</i></code>	Invite player to trade.
<code>/trade_accept</code>	Accepts an offer to trade. Not validated .
<code>/trade_decline</code>	Declines an offer to trade. Not validated .
<code>/tray</code>	Toggles the tray window. (Synonyms: <code>toggle tray</code> , <code>window_toggle tray</code>) (Probable bug synonyms: <code>toggle powers</code> , <code>window_toggle powers</code>)
It is not clear exactly what these three commands do; function seems erratic and dependent on existing tray state.	
<code>/traysticky {0-2} {0 1}</code>	Sets the sticky-state of the specified tray (or tray window). 0 is not sticky, any nonzero value is sticky.
<code>/traystickyalt {0-2} {0 1}</code>	Sets the sticky-state of the secondary (or specified?) tray. 0 is not sticky, any nonzero value is sticky. Invalid command?
<code>/traystickyalt2 {0-2} {0 1}</code>	Sets the sticky-state of the tertiary (or specified?) tray. 0 is not sticky, any nonzero value is sticky.
<code>/turnleft</code>	Rotate left a fixed number of degrees (set by <code>speed_turn</code>).

<code>/turnright</code>	Rotate right a fixed number of degrees (set by <code>speed_turn</code>).
<code>/unbind <i>keyname</i></code>	Unbinds a user-bound key and restores it to the default bind. To unbind a key without restoring the default, use <code>/bind <key> "nop"</code> (See also <code>unbind_all</code> and <code>nop</code>)
<code>/unbind_all</code>	Unbinds all user-bound keys and restores them to the default binds. See also <code>unbind</code> , <code>keybind_reset</code> . Use with caution!
<code>/unex</code>	See <code>unsidekick</code>
<code>/unexemplar</code>	See <code>unsidekick</code>
<code>/unfriend <i>name</i></code>	Remove player from friend list. (Synonym: <code>estrangle</code>)
<code>/unhide</code>	Stop hiding from other users in the "who's on" lists. Has been disconnected from <code>ghide/gunhide</code> .
<code>/unhideall</code>	Stop hiding your name from other users in all of the "who's on" lists.
<code>/unhidefriends</code>	Stop hiding your name from other users in the Friends list.
<code>/unhidegchannels</code>	Stop hiding your name from other users in the Global Channels list.
<code>/unhidegfriends</code>	Stop hiding your name from other users in the Global Friends list.
<code>/unhideinvite</code>	Stop hiding your name from other users in invites.
<code>/unhidesearch</code>	Stop hiding your name from other users in search.
<code>/unhidesg</code>	Stop hiding your name from other users in the Supergroup list.
<code>/unhidetell</code>	Stop hiding your name from other users in tells (direct chat).
<code>/unignore <i>name</i></code>	Stop ignoring user.
<code>/unlackey</code>	No longer be a lackey. (Synonyms: <code>unlk</code>)
<code>/unlevelingpact</code>	Bring up dialog for quitting a leveling pact.
<code>/unlk</code>	See <code>unlackey</code>
<code>/unmal</code>	See <code>unmalefactor</code>
<code>/unmalefactor</code>	No longer be a malefactor. (Synonyms: <code>unmal</code>)
<code>/unrsk</code>	See <code>unsidekick</code>
<code>/unselect</code>	Unselects currently selected thing. Bound to ESC by default.
<code>/unsidekick</code>	No longer mentor (or be a sidekick). (Synonyms: <code>unex</code> , <code>unexemplar</code> , <code>unrsk</code> , <code>unsk</code>)
<code>/unsk</code>	See <code>unsidekick</code>
<code>/up</code>	Jump or fly up. Bound to SPACE by default.
<code>/usebumpmaps {0 1}</code>	Use bumpmaps if available. Function unclear .
<code>/usedof {0 1}</code>	Use Depth of Field (DOF) effects if available. Warning: enabling DOF can seriously impact rendering speed and framerate.
<code>/usefp {0 1}</code>	Use floating point render target for HDR effects if available. Function unclear; default seems to be 1
<code>/usehdr {0 1}</code>	Use HDR lighting effects (bloom, tonemapping) if available. Function unclear; default seems to be 0 .
<code>/userenderscale {0 1}</code>	Use renderscaling if available; see also <code>renderscale</code> , <code>renderscalex</code> , <code>renderscaley</code> , <code>rendersize</code> .
<code>/usewater {0 1 2}</code>	Use fancy water effects if available.

<code>/vis_scale {0.0-4.0}</code>	Controls draw distance. 1.0=default. Set closer to improve performance, further to improve your alt's visual acuity. Up to 2.0 does not seem to impact framerate much; higher settings have notable impact. Experiment with very low settings for amusement.
<code>/watching</code>	List all channels that you belong to.
<i>See Appendix G for complete information on using window save and load functions.</i>	
<code>/wdw_load</code>	Reads window configuration from the file wdw.txt in the default installation folder.
<code>/wdw_load_file filename</code>	Reads window configuration from the specified filename in the default installation folder, or, if specified, in a different location.
<code>/wdw_save</code>	Saves window configuration to the file wdw.txt in the default installation folder.
<code>/wdw_save_file filename</code>	Saves window configuration to the specified filename in the default installation folder, or, if specified, in a different location.
<code>/whereami</code>	Tells you mission name, map name and location.
<code>/whisper name, message_string</code>	See <code>tell</code>
<code>/who name</code>	Show info on player. Appears to be identical to <code>search</code> except for requiring full or partial name string.
<code>/whoall</code>	List who's on the current map, in the system chat window.
<code>/windowcloseextra</code>	Reset UI by leaving in fullscreen mode, closing dialogs and closing all secondary (nonessential) windows. (Same as <code>gamereturn</code>)
<code>/window_color R G B T</code>	Changes the window colors. R-G-B-T should each be replaced with a number from 0-255, where R=Red, G=Green, B=Blue and T=Transparency percentage.
<code>/window_hide window_name</code>	Forces the given window to be hidden.
<code>/window_resetall</code>	Resets all window locations, sizes, and visibility to their defaults.
<code>/window_scale window_name {0.6-2.0}</code>	Changes the named window to the display scale indicated. Range limits shown appear to be the maximum; some displays may have a smaller allowable range.
<code>/window_show window_name</code>	Forces the given window to be shown. (Synonym: <code>show</code>)
<code>/window_toggle window_name</code>	Show a window if hidden, hide a window if shown. (Synonym: <code>toggle</code>)
<code>/y message_string</code>	See <code>broadcast</code>
<code>/yell message_string</code>	See <code>broadcast</code>
<code>/zoomin {0 1}</code>	Controls the zooming in of the view, along with <code>zoomout</code> . Usually used with the <code>+</code> and <code>++</code> modifiers. (As with the <code>lookup/lookdown</code> pair, this command pair will accept the 0/1 variable: if both are set to 1 or 0, camera zooming is unaffected; if one or the other is set to 1, the zoom will persist towards one extreme. There might be some useful purpose to this, but it's eluded me. I think it's a slightly buggy side effect.)
<code>/zoomout {0 1}</code>	See <code>zoomin</code>

Appendix B: Group List of Slash Commands

Slash commands listed by functional group. Refer to the prior section for details of use. Synonyms are separated by commas. Commands may appear in more than one group as appropriate.

System

bug
buy_coh
cmdlist
copydebuginfo
gmotd
help
localtime
logchat
neterrorcorrection
netgraph
noreport
petition
quit
release
requestexitmission
servertime
showfps
stuck
sync, synch
texwordeditor

Viewpoint Control

camdist
camdistadjust
camreset
camrotate
camturn
canlook
face
first
lookdown
lookup
mouse_invert
mouse_look
mouse_speed
~~mouse_pitch_mode~~
playerturn
third
zoomin
zoomout

Movement

autorun
backward
clicktomove, ctm
ctm_invert
ctm_toggle

down
face
follow
forward
forward_mouse
left
mouse_invert
mouse_look
mousepitchmode
mouse_speed
nojumprepeat
playerturn
right
speed_turn
turnleft
turnright
up

Character Control

afk
cc, costume_change
cc_e, cc_emote
change_handle
info_self
info_self_tab
myhandle
quit
release
respec
requestexitmission
set_title
stuck
suppressclosefx
suppressclosefxdist

Friends & Teampay

assist, assist_name
buffs
estrange, unfriend
ex, exemplar
f
fl, friendlist
friend
g, group
getlocalinvite
gfriend
gfriend_player
gfriends

ginvite
ginvite_sg
gunfriend
gunfriend_player
i, invite
k, kick
leaveteam
levelingpact
lfg
lfgset
lk, lackey (CoV)
lp
ml, makeleader
mal, malefactor (CoV)
playernote
playernotelocal
rsk, sk, sidekick
sidekickaccept
sidekickdecline
team
team_accept
team_decline
team_kick_internal
team_quit_internal
team_select
team_task
unex, unexemplar
unlk, unlackey
unmal, unmalefactor
unrsk, unsk, unsidekick

Search & Information

comment
fl, friendlist
friend
getcomment
getglobalname
getglobalsilent
getlocalname
getpos, loc
gfriends
ghide
gignore
gignoring
gunhide
gunignore
hide
hideall

hidefriends
 hidegchannels
 hidegfriends
 hideinvite
 hidesearch
 hidesg
 hidetell
 ignore
 ignorespammer
 ignorelist
 info
 info_self
 info_self_tab
 info_tab
 kiosk
 link_channel
 link_info
 link_interact
 link_interact_global
 map
 myhandle
 netgraph
 newspaper
 playernote
 playernotelocal
 sea, search, findmember
 unfriend
 unhide
 unhideall
 unhidefriends
 unhidegchannels
 unhidegfriends
 unhideinvite
 unhidesearch
 unhidesg
 unhidetell
 unhideprimarychat
 unignore
 whereami
 who
 whoall

Chat & Channels

ac, arena
 b, broadcast, y, yell
 auction, req, request, sell
 autoreply
 beginchat
 c, coalition
 chan_create
 chan_desc
 chan_invite
 chan_invite_gf

chan_invite_sg
 chan_invite_team
 chan_join
 chan_leave
 chan_members
 chan_mode
 chan_motd
 chan_send
 chan_user_mode
 chat
 chat_beta
 chat_cycle
 chat_load
 chat_load_file
 chat_save
 chat_save_file
 chat_set
 chatoptions
 clearchat
 copychat
 e, em, emote, me
 f
 g, group
 guide, h, hc, helpchat
 l, local
 lfg
 lfgset
 logchat
 quickchat
 r, reply
 s, say
 send
 sg, supergroup
 slashchat
 startchat
 t, tell, p, private, whisper
 team
 tl, tell_last
 trade
 trade_accept
 trade_decline
 watching

Global Chat

gfriend
 gfriends
 gfriend_player
 ghide
 gignore
 gignoring
 ginvite
 ginvite_sg

gunfriend
 gunhide
 gunignore
 link_channel
link_info

UI Control

alttray
 alt2tray
 alttraysticky
 buffs
 chat
 chat_cycle
 chatoptions
 clearchat
 clear_reward_choice
 clear_tray
 contextmenu
 dialog_answer
 dialog_no
 dialog_yes
 gamereturn, windowcloseextra
 gmotd
 goto_tray
 goto_tray_alt
 goto_tray_alt2
 goto_trays_tray
 hideprimarychat
 manage
 map
 maximize
 menu
 mouse_invert
 mouse_look
 mouse_look
 mouse_speed
 nav
 next_tray
 next_tray_alt
 next_tray_alt2
 next_trays_tray
 popmenu
 powers
 prev_tray
 prev_tray_alt
 prev_tray_alt2
 prev_trays_tray
 screen
 show
 shownewtray
 tabglobalnext
 tabglobalprev
 tabnext

tabprev
tabselect
tabtoggle
toggle
tray
tray_sticky
tray_sticky_alt
tray_sticky_alt2
windowcolor
window_hide
window_resetall
window_scale
window_show
window_toggle

UI Graphics Control

autoperf
bloomscale
bloomweight
compatiblecursors
cursorcache
disable2D
dofweight
fsaa
fullscreen
~~lightmaplodscale~~
~~ledbias~~
maxfps
maxinactivefps
netgraph
norenderthread
nosunflare
option_set
option_toggle
reloadgfx
renderscale
renderscalefilter
renderscalex
renderscaley
rendersize
renderthread
shaderdetail
shadowvol
showfps
ss
stopinactivedisplay
suppressclosefx
suppressclosefxdist
texaniso
texwordeditor
usebumpmaps
usedof
usefp

usehdr
userenderscale
usewater
vis_scale

Power Execution

insp_combine
insp_delete
inspexec_name
inspexec_pet_name
inspexec_pet_target
inspexec_slot
inspexec_tray
inspirationslot
powexec_abort
powexec_altslot
powexec_alt2slot
powexec_auto
powexec_name
powexec_slot
powexec_toggleoff
powexec_toggleon
powexec_tray
powexec_unqueue

Targeting

target
target_custom_far
target_custom_near
target_custom_next
target_custom_prev
target_enemy_far
target_enemy_near
target_enemy_next,
toggle_enemy
target_enemy_prev,
toggle_enemy_prev
target_friend_far
target_friend_near
target_friend_next
target_friend_prev
unselect

Pets

clear_petnames
petcom
petcom_all
petcom_name
petcom_pow
petoptions
petrename
petrename_name
petsay
petsay_all

petsay_name
petsay_pow
pet_select
pet_select_name
release_pets
showpetnames

Arena

ac, arena
ai, arenainvite
getarenastats
getallarenastats
getratedarenastats

Screenshots & Recording

demorecord
demostop
e3screenshot
profiler_record
profiler_stop
screenshot
screenshottga
screenshottitle
screenshotui

Bind & Macro

bind
bind_load
bind_load_file
bind_load_file_silent
bind_save
bind_save_file
bind_save_file_silent
keybind_reset
macro
macroslot
nop
unbind
unbind_all

SG & Coalition

c, coalition
coalition_cancel
ci, coalition_invite
coalition_mintalkrank
coalition_nosend
coalition_sg_mintalkrank
demote
ginvite_sg
namecaptain
namecommander (CoV)
nameenforcer (CoV)
nameflunky (CoV)

nameleader
namelieutenant
namemember
nameoverlord (CoV)
nameringleader
nametaskmaster (CoV)
promote
raid_invite
sg, supergroup
sgcreate
sgi, sginvite
sgk, sgkick
sgkickyes
sgleave
sgmode
sgmodeset
sgraidinvite
sgraidwindow

sgsetcostume
sgsetdemotetimeout
sgsetdescription
sgsetmotd
sgsetmotto
sgstats

Email

emaildelete
emailheaders
emailread
emailsend

UI Save & Restore

chat_load
chat_load_file
chat_save
chat_save_file

option_load
option_load_file
option_save
option_save_file
option_set
option_toggle
wdw_load
wdw_load_file
wdw_save
wdw_save_file

Architect/Unknown

clear_attribute_view
mission_make
mission_search
mm_entry
monitor_attribute

Appendix C: Bindable Key Names

Unless noted, all keys can be bound with the ALT+, CTRL+ and SHIFT+ modifiers.

You should avoid changing or attempting binds on the middle mouse button and wheel – the default binds appear to be persistent and it is difficult to get new binds to be successful.

Key	Notes
A through Z	Main keyboard alphabetical keys. These keys are case-insensitive in bind definitions; F and f are the same key. Use SHIFT+ to bind two commands to the same alpha key.
1 through 0	Top numeric keys. Each of the symbols above the numbers is bindable as SHIFT+[number].
F1 through F12	Top function keys.
SPACE	Space bar.
COMMA	
/ (foreslash)	The ? key is bindable as SHIFT+ /
\ (backslash)	The key is bindable as SHIFT+ \
; (semicolon)	
` (apostrophe)	
- (minus)	
[(left bracket)	The { and } keys are bindable as SHIFT+[and SHIFT+].
] (right bracket)	
BACKSPACE	
END	
ESC	
ENTER	
EQUALS	= key. The + key is bindable as SHIFT+ =.
HOME	
INSERT	Does not appear to be (re)bindable.
LALT	Left ALT key
RALT	Right ALT key
PAGEUP	
PAGEDOWN	
TAB	
SYSRQ	SysReq/PrintScrn key. ALT+SYSRQ not functional.
DELETE	
NUMPAD0 – NUMPAD9	The numeric keypad number keys.
NUMPADENTER	The numeric keypad ENTER key.
DECIMAL	The numeric keypad Del/. key.
MULTIPLY	The numeric keypad * (asterisk/multiply) key.
DIVIDE	The numeric keypad / (slash/divide) key.
SUBTRACT	The numeric keypad - (minus) key.
ADD	The numeric keypad + (plus) key.
UP	Up arrow key.

Key	Notes
DOWN	Down arrow key.
RIGHT	Right arrow key.
LEFT	Left arrow key.
LBUTTON	Left mouse button. Unmodified click is bound to 'select'. See below for some notes on this.
RBUTTON	Right mouse button.
MBUTTON	Middle mouse button. Rebindable only with limitations if a mousewheel is present – very difficult to make clicks work without moving wheel and causing odd reactions.
MOUSECHORD	Combination of the left and right mouse keys. One cool use for this is to bind it to TP, so that as you're running along, steering with the mouse, you can jump over obstacles one-handed.
MOUSEWHEEL	Mouse wheel – does not appear to be rebindable.

Note that the left mouse button can be used in an "instant action" mode, since it is bound to the required 'select' function. Any bind that's bound to this key, or any modified key (CTRL +, ALT +, etc.) will be executed simultaneously with the click. In other words, if you bind a buff command to CTRL+LBUTTON, ctrl-clicking a mate will instantly apply the buff power. Ditto for teleporting – if you bind teleport to a mouse button-key combo, you TP to the clicked spot. Since select can fail (you might click on an invalid object or player, or an invalid TP spot) these kinds of binds can occasionally fail or cause unpredictable actions. But it's a very useful trick and has transformed my Teleport and Recall Friend functions on alts that have that power.

Appendix D: Window & Menu Names

Anyone who knows more about window and menu names is invited to supply the information.

There does not appear to be a keyword applicable to the "Menu" window – the thing with the health bars, XP status and main menu items on it

Keywords in **bold blue** can be used directly to toggle windows (/chat and /map, for example); the others can be used only as command arguments.

Name	Description
badge	Window listing badges
chat	Entire chat window pane
clues	Window listing clues (and souvenirs)
combatnumbers	Combat (damage, buff, defense, etc.) numbers
contact	Window listing contacts
costume	Costume selection window
email	Email management window
enhancements	Tray of acquired/unused enhancements
friend friends	Friends list window (command is different)
help	Help menu
insp	Inspirations tray
manage	Power/slot/enhancement management window – this keyword does not work with the various window-handling commands.
map	Map window
menu	The main menu (NOT the menu-bar element!)
mission	Mission list window
missionreview	Architect mission review window
mm	Architect manager window?
nav compass	Nav window (compass, current mission data)
pet	Pet management window.
playernote	Player notes window (command is different)
power tray	Power trays.
powerlist powers	Powers inventory window
recipe recipes	Recipe inventory window
salvage	Salvage list window

Name	Description
sea search	Team search window
supergroup	Supergroup management window (command is different)
target	Currently selected target window
team group	Team member window (command is different)

Appendix E: Emote Codes

These codes can be executed at almost any time using the slash code `/emote`, `/em` or (amusingly) `/me`. They can also be selected from the QuickChat menu, which is raised by clicking the small button at the right end of the chat text entry window, or by the slash code `/quickchat`. The best way to see what each emote does is to find a quiet corner of the map, use camera rotate (default: PAGEDOWN plus the mouse) to spin around so you're looking at your character from the front, and try each one out.

I have combined some codes out of alphabetical order for clarity.

A code is "static" if it stays until an interrupt key (such as movement) is pressed. Mouselook can often be used during a static emote without interrupting the emote. Powers on auto (like Hasten) can interrupt as well.

Note that many of the QuickChat options are similarly named, but include fixed chat bubbles as well.

If you use any emote string besides one of these valid codes, the string will appear in a thought bubble over your head, visible to others, preceded by your character name. ("Shenanigunner wishes he had a beer.")

Code	Static?	Animation/Artifact	Notes & Description
afk newspaper	Yes	Animated read newspaper	Good "I'm waiting" or AFK emote.
afraid cower fear scared	Yes	Cower in fear	You too can be a civilian.
alakazam	No	Makes dramatic magician gesture	
alakazamreact	No	Try it and see. You wouldn't believe me if I told you.	A major hoot. Just try it.
akimbo wings	Yes	Stand with hands on hips	
angry	No	Animated anger	
assumepositionwall	Yes	Stand against wall as if to be searched	Looks pretty stupid unless you stand facing a wall or other surface as closely as you can before executing.
atease	Yes	Animated stand at ease	
attack	No	Animated one-arm motion	
backflip flip	No	Perform a backflip	
batsmash	Yes	Animated lay about you with a baseball bat	
batsmashreact	Yes	Animated react to getting hit with a baseball bat	

Code	Static?	Animation/Artifact	Notes & Description
bb boombox dropboombox	Yes	Character places boombox in front of him/her	The basic socialization, showoff and time waster emote – haul out the boombox and dance. The tune will be randomly selected from those listed below
bbAltitude bbBeat bbCatchMe bbDance bbDiscoFreak bbDogWalk bbElectroVibe bbHeavyDude bbInfoOverload bbJumpy bbKickIt bbLooker bbMeaty bbMoveOn bbNotorious bbPeace bbQuickie bbRaver bbShuffle bbSpaz bbTechnoid bbVenus bbWahWah bbWindItUp bbYellow	Yes	Boombox + dance	Using these codes will select specific boombox tunes instead of randomly choosing one of them. Avoid newbie zones where as many newbies as possible attempt to set up competing-tune boomboxes. For one thing, it can crash your client. For another, it can crash your brain. See also drumdance.
beatchest tarzan	No	Animated chest-pounding	Audible growl.
biglaugh laugh2 laughtoo	No	Hearty laugh	
bigwave overhere	No	Animated big wave	
binoculars	Yes	Look through binoculars	
blankfiller	?	None?	Appears to be the emote equivalent of “nop” for slash commands. Does nothing but generates no error either.
bow	No	Animated bow	
bowdown	No	Animated demand person before you bow down	
burp	No	Animated burp with sound	Audible over Local distance. Look, I'm a <i>rude</i> Warwolf!

Code	Static?	Animation/Artifact	Notes & Description
buzzoff goaway	No	Animated shooing motion with hand.	
camera	Yes	Take pictures with old-fashioned Speed Graphic camera	Cannot be interrupted until first part of animation is completed.
cellphone	Yes	Talk on cel phone	Wow, that's a big phone!
champion	No	Clasped-hands victory shake	See also xkevqt {.
cheer	Yes	Animated shake-fists encouragement	
chicken	No	Do the chicken dance	
clap	No	Animated clap with sound	Audible over Local distance.
clipboard	Yes	Write on clipboard	
cointoss coin flipcoin	Yes	Animated coinflip motion; show head or tail coin overhead	Make a choice for the team or group, or yourself. Coin remains until interrupted.
crack knuckle knuckles	No	Crack knuckles	Loud sound effects!
crossarms	Yes	Cross arms	
crouch	Yes	Crouch down, frog style	
curseyou noooo	No	Animated shaking fist at the heavens in dismay	
dance	Yes	Animated dancing	Several random dances; repeat emote for others. You can also use the new <i>dancen</i> commands to select specific dances. See also <i>drumdance</i> .
dance1	Yes	Cha-cha dance	
dance2	Yes	Rah-rah dance	
dance3	Yes	Twist dance	
dance4	Yes	Hands waving in air dance	
dance5	Yes	Hands in air hop dance	
dance6	Yes	High-energy twist dance	

Code	Static?	Animation/Artifact	Notes & Description
dice rolldice dice7*	No	Animated dice roll motion; show die overhead	Make a choice for the team or group, or yourself. Die fades after a few seconds. dice7 is a special emote unlocked by completing the Hess task force; the die always rolls 7. Heroes only, although dice cheating would seem to be more appropriate to Villains...
disagree	No	Animated "no" wave with short lecture animation	
dontattack	No	Animated two-hand no	
donut eatdonut	Yes	Eat a donut	
drat	No	Animated thump both fists	Express frustration in a friendly way.
drink	Yes	Drink from glass	See also eat, donut and teabag.
drum	Yes	Pound on huge tribal drum	Loud sound effects.
drumdance	Yes	Animated little "raindance"	See also bb and dance.
drumlow	Yes	Pound on small tribal drum	Bongo sound effects.
dustoff	No	Brush off hands	
eat food	Yes	Eat food item	Alternates between burger, hot dog and sandwich.
evillaugh elaugh muahahaha villainlaugh villainouslaugh	No	Animated "bwah-ha-hah" villain laugh	(How many synonyms are needed for one emote!?)
explain	No	Animated "hold it," with short lecture animation	Cannot be interrupted until first part of animation is completed. See also lecture.
fancybow elegantbow	No	Animated elaborate bow	
fishing	Yes	Fish with long pole	
flashlight flashlightdown	Yes	Animated looking around with large flashlight over shoulder, pointed mid-downwards	Does not appear to project light.

Code	Static?	Animation/Artifact	Notes & Description
flashlightup	Yes	Animated looking around with large flashlight over shoulder, pointed mid-upwards	Does not appear to project light.
flex flex1 -or- flexa flex2 -or- flexb flex3 -or- flexc	Yes	Animated bodybuilder poses	Impress newbies and that cute controller by doing your Arnie impression. Three different poses for your convenience. (flex and flex2 are the same.)
flippingcoin	Yes	Flip coin gambler style	Not same as flip or flipcoin; does not generate "result."
floatbooks	Yes	Float three books in front of you and appear to study them	
These four emotes work only when you are already flying. If you pause, your character will revert to the standard flying pose. There is no emote to return to the standard flying posture. A fly-forward plus emote keybind is recommended for regular use, or a keybind that cycles through the options.			
flypose1	Yes	Fly with fists out front	
flypose2	Yes	Fly with one fist out front	Superman pose
flypose3	Yes	Fly with hands flat out front	Swan dive pose
flypose4	Yes	Fly with fists to sides	Invisible hang glider pose?
frustrated	Yes	Animated shake both fists	Stays in fist-clenched posture after shake.
getsome kissit	No	Turn fanny to front, pat it	Ruuuuude. Love it.
grief	Yes	Animated grief on knees	Stays on knees after initial animation.
hand talktohand	No	Animated hand out in "talk to the hand!" style	Yeah, right, enough outta you.
handsup surrender	Yes	Animated hands in the air	Alternate positions: standing and kneeling
hi wave	No	Animated wave	
holdtorch	Yes	Animated hold a tall flaming torch	Not sure if it actually projects any light.
howl	No	Howl like a warwolf	Great sound effects.
hmmm plotting	No	Stare into space and rub chin	

Code	Static?	Animation/Artifact	Notes & Description
huh shrug what	No	Animated shrug	
invent	Yes	Manipulate a cool luminescent grid thingy	That or it's a new-gen Rubik's Cube. Used whenever a character is interacting with an invention table.
jumpingjacks	Yes	Animated jumping jacks	With sound effects.
kata martialarts	Yes	Animated kata (martial arts practice moves)	
kneel	Yes	Animated kneel	
laptop	Yes	Animated work on laptop that appears on pedestal	Occasionally seem to experience computer trouble. (Is this a backhanded joke at a "boss" key?) See also <code>type</code> .
laugh	No	Animated hands-on-hips laugh	Why, yes, I am Errol Flynn!
lecture	No	Animated lecturing poses	
ledgesit	Yes	Animated sit on edge (of wall, building, etc.)	Alternate positions: hands on surface and hands on knees. Legs will disappear into ground if done anywhere but facing outward on a ledge. See also <code>sit</code> and <code>walllean</code> .
listenpoliceband	Yes	Whip out your way-cool holographic police radio	Used for the police band mission contact. Hero side only.
lotus yoga	Yes	Animated lotus position	Sophisticated resting posture.
marriageproposal*	Yes	Down-on-knee proposal	Only available with Wedding Pack.
militarysalute	Yes	Animated formal salute pose	Static pose, unlike <code>salute</code> .
newspaper			See <code>afk</code> .
no	No	Animated wave-hands "no"	See also <code>disagree</code> .
nod	No	Animated nod	
panhandle	Yes	Animated sit with cup, offering as to passersby, occasionally looking it it disappointedly	One way to bug inf off of high-level players.
paper			See <code>rock</code> .

Code	Static?	Animation/Artifact	Notes & Description
peerin	Yes	Animated peering in window with hands cupped around face	Occasional look-around to see who's watching.
plot scheme	Yes	Hunch and rub hands together as if making evil scheme	
point	No	Animated one-hand point straight ahead	
praise	Yes	Animated salaam on knees	
protest	Yes	Animated shaking large protest sign	Appear to be three different signs that come up at random. All are illegible except for a large STOP, NO and red circle/slash over an indistinct outline of something. Must halt emote to change signs.
raisehand stop	Yes	Animated raise one hand	
readbook	Yes	Read from book	
research	Yes	Animated refer to book, then examine what's in front of you	Circle of Thorns seen doing this in Hollows and elsewhere.
researchlow	Yes	Animated refer to book, then examine what's in front of you, while squatting down	Circle of Thorns seen doing this in Hollows and elsewhere.
roar	No	Animated roar with sound	Look, I'm a Warwolf!
rock scissors paper	Yes	Play Rock-Scissors-Paper (aka Rochambeau) with animation	Settle disputes. Animation shows all three icons for five seconds, then your selected one.
rooting wavefist	No	Animated wave fist, hands-to-face shout and clap with sound	Only clapping has sound.
salute	No	Animated casual salute	Use <code>militarysalute</code> for a formal, held salute.
score1 score2 ... score9 score10	Yes	Hold up score card with 1 to 10 on it, Olympics-style	Show your opinion of another player's move. Fun to use with costume contests, etc. (What, no score zero?)
screen touchglass	Yes	Reach out and touch surface in front of you as if not sure it's there, or touch wall-screens	Fabulous animation if character is in a bubble or if you pivot viewpoint so that you're looking right into character's face. Fun.

Code	Static?	Animation/Artifact	Notes & Description
shucks	No	Animated thump one fist	Aw, it was nothing.
sit	Yes	Animated sit down	Take a load off. Fun to do on benches, trees, etc. See also <code>ledgesit</code> and the other <code>sit...</code> commands.
Advanced sit commands		See end of Appendix	
slap	No	Animated forehand slap	With light burst and slap sound. Combine with <code>slapreact</code> from other character for more fun.
slapreact	No	Animated reaction to being slapped or struck	
slash slashthroat	No	Animated draw finger across throat.	Stop; Shut up, dude; <i>or</i> You're dead, you know.
sleep	Yes	Fall asleep standing up, with stream of Z's rising	
smack	No	Animated backhand slap	Great sound effect. See <code>slap</code> .
smackyou threathand	No	Threaten to backhand someone	See <code>smack</code> .
sorry	No	Animated apology gesture	
talk	Yes	Talk as if in conversation	
taunt taunt2 <i>-or-</i> tauntb	No	Animated two-hand taunt with "hoooah" sound	Character stays in combat pose after taunt
taunt1 <i>-or-</i> taunta	Yes	Animated one-hand taunt with "aaaaah" sound	Character continues to pound fists with sound effect after taunt
teabag	Yes	Dunk teabag in a teacup	See also <code>drink</code> , <code>eat</code> .
thanks thankyou	No	Animated left-hand gesture	See also <code>yourewelcome</code> , which is a mirror-image gesture.
thewave	No	Vertical "wave" animation	
throwconfetti* throwrice* throwrosepetals*	Yes	Throw confetti, rice or rose petals	Only available with Wedding Pack.
thumbsup yes	No	Thumbs-up animation with nod	

Code	Static?	Animation/Artifact	Notes & Description
type typing	Yes	Type as if on keyboard – same as laptop but without prop	Great for consoles in missions.
victory	No	Animated victory arm wave	See also champion.
waiting	Yes	Various impatient waiting actions	
walllean	Yes	Animated relaxed lean back against wall	Alternate positions: hands in pockets or arms crossed. Stand as close to wall or object as possible before executing. You can also get into amusing positions if you do it back-to-back with static NPCs – it looks as if your character and the NPC are in a <i>very</i> close embrace.
welcome	No	Animated two-hand welcome	
whistle	No	Animated one-hand whistle with piercing sound	Audible over Local distance – loud!
winner	No	Animated clasped-fist victory wave	
wounded	Yes	Wobble woozily	Like a weebelo. Remember Weebelos?
yatayata yata	No	Animated “talk-talk- talk” with hand	
yourewelcome	No	Animated right-hand gesture	See also thanks, which is a mirror-image gesture.

Advanced Sit Emotes

A huge selection of fancy sit emotes was added with Issue 8. They are somewhat complicated to list, because they are different for male/huge and female characters. (Ladies sit differently, guys, in case you've never noticed...)

All are static.

Note: I have not tested these with Huge characters. I assume they are the same as male but if someone wants to test things and report back...

Emote	Male & Huge action	Female action
sitbench1	Legs out straight, hands straight behind	Same
sitbench2	Sideways sprawl with one leg up and one arm along bench back	Same as sitchair1
sitbench3	Sprawled back, feet flat, arms on bench back	Same as sitchair1
sitbench4	Same as sitchair3	Same as sitchair1 – elevates over surfaces, though.
sitchair1	Straight back, feet flat, hands on knees	Straight back, knees crossed, hands center
sitchair2	Leaning forward, feet flat, hands loose in middle	Leaning back, feet flat, hands on thighs
sitchair3	Straight back, feet flat, hands on thighs	Same as sitchair1
sitexecutivechair	Lean back, hands on chair arms, feet flat	Same but legs crossed
sitstool	1 foot down, 1 foot on rungs, 1 hand on knee	Feet up on rungs, legs crossed, hands clasped on knee
sittable1	Straight back, knees loose, 1 arm on table, other hand to face	Same but knees together
sittable2	Same as sittable1, but hands loose on table	Same as sittable1, but lower table surface

Appendix F: Chat Bubble Color Codes

It is possible to change the appearance of your character's chat bubble in two ways. The simplest is to set the text color and the background color in the Options menu. For some reason, though, this setting only affects some chat bubbles; many will default to black-on-white.

The second way to set chat bubble color – and other characteristics – is to use inline format codes. These codes can be used in manually entered chat strings or as parts of binds. The complete code set is:

```
<color ccode><bgcolor ccodetransparency><border ccode><scale factor><duration seconds>
```

As far as I know, each command can be used separately and in any order.

`color` sets the text color. The value `ccode` can be any standard color name (not sure of the range, but basics like red, yellow, white, blue etc. should all work). You can also use hex codes in the `#rrggbb` format – look up those codes anywhere on the web if you're not familiar with them. This works the same as the text slider in the Options menu.

`bgcolor` sets the chat bubble background color, and works the same as `color` except that you can add an additional value to control the chat bubble background transparency. If you use only a color code, you get 100% color (that is, no transparency). If you add two digits to the end of the color code, you set the transparency, from 0 to 99%, with zero being fully transparent. This setting does not appear to have full 100-step granularity; there may be as few as 8 steps of transparency. I am not sure if strings like `'yellow50'` will work, but codes like `'#FFFF0050'` will.

`border` sets the color of the bubble border. Identical in operation to `color`.

`scale` sets the text and bubble size. It is supposed to scale from 0.0 to 4.0, with 1.0 being the default size, but it only works 0-2.0 for me. Useful for blowing up important bubbles like "Here!" when you've found more foes or a glowie.

`duration` sets the persistence of the bubble in seconds. Default is about 8 seconds. You can make bubbles like "Here!" more persistent, to give mates time to find you, by setting the value to 15 or so.

To use this method, embed the codes in a chat string, like this simple example:

```
g <color red><bgcolor black>Oh, no, dead again!
```

Note that any spaces between the codes will be added to the chat string.

If you want to make all your chat bubbles a specific style, or have multiple styles for different uses, you need to bind a key to start the chat and load the codes – then you type your message after the codes. A little murky, but it works. For example, the normal Chat key is Enter, so:

```
/bind ENTER "beginchat <color white><bgcolor blue><scale 2><duration 10>"
```

And whenever you press ENTER, you'll be ready to chat in large white-on-blue text with a 10-second persistence. The same thing can be used in general binds:

```
/bind CTRL+T "g <color blue><bgcolor red>Teleporting $target to  
me!$$powexecname Recall Friend"
```

...although be warned I have found this usage to get flaky at times.

A final bind you might find useful to experiment with or frequently change the settings is:

```
/bind CTRL+F1 "beginchat /bind ENTER "<color #00000><bgcolor #FFFFFF'75>  
<border #FF0000><scale 1.0><duration 10>"
```

This mess will, when you press CTRL+F1, load the chat entry window with "/bind..." and the whole string that follows. Edit it to suit, press ENTER, and then use ENTER to start new chat lines with the edited characteristics. You've changed your keybind for ENTER by doing so. This can create a complete mess if you're not careful, so... be careful.

Appendix G: Saving & Loading Interface Settings

With Issue 11 or 12, City of Heroes/Villains finally resolved one of the most annoying oversights in its design. Each new alt that you designed started with a generic user interface setup, and there was no way to duplicate a favorite layout and setup without laboriously configuring each element, every time. Now, however, there are not one but three separate “save/load” functions to save an aspect of a customized user interface and reload it into another character’s interface.

There are three sets of customization commands, for chat, window layout, and the grab-bag “options.” All work much like the process for saving and loading binds and macros, so any user who has mastered those basics should have no trouble with these facilities.

Chat Configuration Save and Load

Saving a carefully designed chat window setup is now trivial.

Step 1 is to set up your chat windows as you like them, down to the last detail, on any of your characters.

Step 2 is to save the chat window configuration. The `chat_save` command will save the chat configuration in the default game folder, in the file `chat.txt`. More sophisticated users will probably want to use `chat_save_file`, which will save the chat configuration to a specified path and file. The latter also permits saving more than one chat configuration.

Step 3, load the new configuration into each character’s interface. The `chat_load` command will load the default `chat.txt` file, while `chat_load_file` can be used to load any filename on any path.

Warning: It may be possible to directly edit the `chat.txt` file, but one look at it showed some complex and cryptic components (like numeric strings that likely reference specific channels). All but the most advanced users are recommended to leave the file contents alone and do all chat configuration from within the user interface.

Window Configuration Save and Load

Saving your individual preference for window layout and arrangement is now trivial.

Step 1 is to set up your user interface windows as you like them, down to the last detail, on any of your characters.

Step 2 is to save the window configuration. The `wdw_save` command will save the window configuration in the default game folder, in the file `wdw.txt`. More sophisticated users will probably want to use `wdw_save_file`, which will save the window configuration to a specified path and file. The latter also permits saving more than one window configuration. (This opens possibilities for fast switching between patrol and combat window layouts, for example, or even several layouts, each optimized for a particular area of gameplay.)

Step 3, load the new window configuration into each character’s interface. The `wdw_load` command will load the default `wdw.txt` file, while `wdw_load_file` can be used to load any filename on any path.

The `wdw.txt` file is probably editable by any savvy user. Even so, most users are recommended to leave the file contents alone and do all window configuration from within the user interface.

Option Configuration

Ah. Now the good stuff – the feature that lets you set any of several dozen game parameters, either individually or by loading a saved file. The Devs decided to call this grab bag “options.”

Simply saving and loading option configuration files is the same as saving and loading bind, chat and window configurations. Let’s summarize that quickly:

Step 1 is to set all of your options in the configuration menu, down to the last detail, on any of your characters.

Step 2 is to save the option configuration. The `option_save` command will save the option configuration in the default game folder, in the file `options.txt`. More sophisticated users will probably want to use `option_save_file`, which will save the option configuration to a specified path and file. The latter also permits saving more than one option configuration.

Step 3, load the new option configuration into each character’s interface. The `option_load` command will load the default `option.txt` file, while `option_load_file` can be used to load any filename on any path.

The option.txt file seems to be readily editable, as the contents are merely the option keywords and the status or values.

You can also set individual options via the slash command option_set, which takes two arguments: the option keyword and the new value. For example, you can toggle on dirty word bleeping with the following command:

```
/optionset allowprofanity 0
```

And return to seeing every word your angry tank wants to type by using:

```
/optionset allowprofanity 1
```

Even simpler, most options can be toggled from one state to the other using optiontoggle:

```
/optiontoggle allowprofanity
```

will simply flip the setting from one state to the other.

Ah, but you ask, what are the available option keywords? Simple: use the command

```
/option_list
```

and the complete, current list of option keywords will scroll past in the chat window. You can use logchat to capture the stream for offline examination.

Here is the list as of initial release of Issue 14. It is assumed that most commands are set by simple 0/1 values. Most should be self-explanatory to a moderately experienced player. At some point I may expand this section with notes on the unusual keyword. For now, I recommend that you save your own options file and examine its contents, and the values associated with non-toggle keywords. Please do send along anything interesting you discover!

AdvancedPetControls	AllowProfanity
ArchitectAutoSave	ArchitectBlockComment
ArchitectNav	ArchitectToolTips
AutoDeclineSuperGroupInvite	AutoDeclineTradeInvite
BuffSettings	CamFree
Chat1Fade	Chat2Fade
Chat3Fade	Chat4Fade
ChatBubbleColor1	ChatBubbleColor2
ChatDisablePetSay	ChatEnablePetTeamSay
ChatFade	CompassFade
ContactSort	DeclineGifts
DeclineGiftsFromTeammates	DefaultChatFontSize
DisableCameraShake	DisableDrag
DisableLoadingTips	DisableMouseScroll
DoNotSeeEnemyLocal	EnableChatLog
EnableClickToMove	EnableJoystick
FadeExtraTrays	gShowPetBuffs
HideButtons	HideEnhancementFullMsg
HideHeader	HideInspirationFullMsg
HidePetNames	HidePromptCoop
HidePromptDeleteEnhancement	HidePromptDeleteRecipe

HidePromptDeleteSalvage	HidePromptPlaceEnhancement
HideRecipeFullMsg	HideSalvageFullMsg
LogPrivateMessages	MapOptions
MouseButtonReverse	MouseInvert
MousePitchSetting	MouseScrollSpeed
MouseSpeed	NoXP
PreventPetIconDrag	PromptTeleportFromTeammates
RecipeHideMissingParts	RecipeHideMissingPartsBench
RecipeHideUnowned	RecipeHideUnownedBench
SeeEnemyBroadcast	ShowArchetype
ShowAssistReticles	ShowBallons
ShowEnemyTells	ShowOwnerName
ShowPetControls	ShowPets
ShowPlayerBars	ShowPlayerName
ShowPlayerRating	ShowPlayerReticles
ShowSupergroup	ShowVillainBars
ShowVillainName	ShowVillainReticles
SpeedTurn	StaticColorsPerName
TeamComplete	ToolTipDelaySec
UseToolTips	WebHideBadges
WebHideFriends	WebHidePowers
WebHideStats	WindowFade

Revision History

- 0.50 18 Feb 2005 First release.
- 0.51 19 Feb 2005 Minor corrections throughout. Deleted inoperative `bindlist` command. Updated `requestexitmission` command. Added three items to Appendix C, Window Names. Completely rewrote and extended Appendix B, Bindable Keys. Thanks to Xocyll for much feedback on this one – and others on small details.
- 1.00 15 Feb 2005 Added Issue 4 and 5 commands and updated others I've been able to verify or learn more details about. I still have no experience with the chat channel stuff – if you have experience, look over the entries and send me updates or confirmation.
- 1.10 01 Dec 2005 Added Issue 6 commands, including quite a few new emotes.
- 1.50 06 Jan 2006 Added quite a few new commands discovered in the command list and updated many of the existing ones. Broke out slash commands into group table as Appendix B.
- 1.60 22 Mar 2006 Updated emotes list, mostly with synonyms. Updated some slash command entries. Updated window list. Updated keyname list. Expanded title and references to CoV.
- 1.70 12 Jun 2006 Issue 7 update. Updated emotes list. Updated slash command entries.
- 1.80 30 Nov 2006 Issue 8 update. Mostly new emotes, plus addition of the chat bubble color codes.
- 2.00 13 April 2009 (Issues 9-14 update.) Wow, getting to this a little late. Fortunately, the command and emote base has stayed relatively stable over the last several releases. This will likely be the last major update of this guide; I hope you've found it useful!

The End.