

## Gunner's Advanced Bindfile System (v1.0.0) — from the Legendary Death-Defying Shenanigunner!

### CONTENTS

- Quick Start & File Setup ————— p.1–2
- GABS Bindfile Reference List ————— p.3–11
- Bindfile Techniques ————— p.12–14
- GABS Bind Reference Chart ————— p.15
- GABS Master Bind Chart ————— p.16

### Install the GABS Files

1. All files for this system should be placed in the game's default folder for user files: **[GAME FOLDER]\settings\live**. Unpack the **GABS.zip** file there. (This assumes you are using the Homecoming Launcher; if you are still using the Tequila Launcher, the folder is **\data**.)

Using the default location is highly recommended because it is where the game saves and loads these files by default. You can use any other path you like but you will have to edit all the files to reflect it, and thus all file references will have to be that much longer. It's useful to make a shortcut to this folder and put it on the desktop, for quicker editing.

2. Make a subfolder named **\GABS** and copy all the GABS component files there (just unpack the ZIP file again) as a backup archive. This is also a good place to put the documentation files and any support files you create. For backup/restart purposes, you might keep the original archive file there, too.

*It is recommended that you edit only the files actually in the **\live** folder, just to preserve backup copies and keep from getting confused. (It's all too easy to edit an archive copy and then wonder why your new commands aren't loading, with increasing degrees of frustration...)*

3. Review the **COMMON.txt** file listing in the reference listing that follows. These are binds and commands that will be used by each one of your alts, and this is the place to make **any changes you want shared among them**. Be sure to keep a copy of the original file at hand for replacement or a fresh start, if needed.

*It is strongly recommended that you make no changes to the **COMMON** binds until you have the GABS system set up for two or three alts and*

*thus have a better understanding of how it works as a shared system. But this is the time to make any small tweaks that might keep the binds from working as desired on your system, or if you have strong preferences about some of my suggested keys.*

4. Copy the **[ALTNAME].txt** file to the name of your first character. (Such as **SooperDoooperMan.txt**.) Use a short, filename-safe version of any long or complex name, to make loading it within the game simpler—such as **SDMan.txt**, with no spaces or special characters.

DON'T edit the original source file, at least, not yet. There may be global changes you want to make for all your alts, but let's get one version completely up and working before you make those choices.

5. Open the new alt file in a plain-text editor (Notepad is fine; I'm a big fan of Notepad++; DreamWeaver or Visual Source Editor are good.)

**Do NOT use Word or any equivalent**—the file has to stay plain ASCII and all quotes have to remain 'straight quotes,' not the curly ones you see in this document. Getting upper-ASCII characters in the file will have disastrous and unpredictable results at testing time.

### Editing Your Alt's Binds

Okay, now the hard part. Or at least the Task Force-grind-part.

1. With the reference pages that follow, go through your new alt bind file and review each bind that's defined or suggested there. Edit each one that is not a good match for your alt, play style, power sets, etc.

This will be an iterative process until you have all desired binds mapped and are done tweaking them... and then, especially if you're new to advanced use of keybinds, the tweaking and experimenting may never end. And then your alt gains new powers, so away you go again.

*So I strongly recommend, especially on this first implementation, that you go very slowly. Don't add any binds to the file other than perhaps one or two simple and obvious ones. Let's get this simple version of everything up and working, and get familiarity with the process, before diving in too deep. I suggest you do only the following:*

- Use search/replace to change every instance of **[ALTNAME]** to that short, filename-safe form of your alt's name, with no spaces or punctuation. (These are all for filenames, and may need to be typed in the game.)
- In the reference list, find every instance of **magenta ("editable")** text and correct the corresponding entry in the alt file. Don't waste too much time on ones you will want to tweak, such as the parameters on **EQUALS**. Focus on functionality, for this first pass.
- In the file, find every instance of a commented-out line (such as **W**, or **Y**) and either edit it to suit your needs or delete the unwanted duplicate.

Save the Alt binds file. (If you made any modifications, make sure the **COMMON** file is saved as well. You can leave them open in the editor.)

2. In the **\live** files, find the two generic "REST" files and duplicate them.
  - Rename the duplicate files to match the name as it was specified in the Alt file edit, from **[ALTNAME]-REST-OFF.txt** to (say) **SDMan-REST-OFF.txt**, and the same for the **-ON** file.
  - Edit both files and change the filename within to match the actual filenames. Notice that the filenames criss-cross!

*The name form of these filenames (in the Alt binds, on the files and in the two files) need to be an exact match for the Rest bind to work! All five file names should be identical EXCEPT for the criss-crossing -ON and -OFF.*

### Testing Bind Changes

For testing bindfile changes, now and all on down the road, it can be done by loading and exiting the game, but you can also keep the editor open and toggle to it, either on some spare monitor space or by Alt-TABbing back and forth. It's very convenient to make fixes and changes, save the file, and then swap to the game to load and test it.

1. Log in to the alt for which you've edited the new bindfile, and take him/her/it/Their Majesty to a quiet, private place.

- You will probably want to go to your SG base, or at least to a safe, unpopulated corner of a map, and set your chat to **Local** or **SG** to avoid sending out annoying garbage for the amusement of other players. Trust me, you want to pay attention to this step. No, don't ask me why.
- Make sure your chat windows are displaying the **System** and **Error** channels, as this is where bind reporting appears. Type a nonsense command like **/fooble**, and you should see yellow Error text; type a command expecting arguments, like **/up**, and you should see pink System text.
- You may even want to create a dedicated **Testing** tab for these.

## 2. Replace your binds:

- First, save your existing binds. Type **/bindsavefile [ALTNAME]oldbinds.txt**, using a short alt name.
- Clear all keybinds: type **/unbind\_all**. This will instantly wipe out any custom binds and reset ALL keyboard and mouse commands to the game defaults. This is one good reason you want to be somewhere safe; you may not have full control of your powers!
- Load the COMMON binds file: type **/bindloadfile COMMON.txt**. You should get a green acknowledgment message. If not, read the yellow error message carefully.
- Load the Alt binds file: type **/bindloadfile [ALTNAME].txt**, using the name form you used for the file. Again look for a green ack message.
- Press **Alt-F5** to save a copy of the complete bindfile for this alt. Now, pressing Alt-F6 will reload what should be a known-good or at least un-mangled bindfile when needed. Try to save the bindfile only when you are certain it has no problems; it can be a real pain to be "trapped" with broken binds!

## Test Your Binds

Yeah, sounds easy. But be sure to test every bind on the main GABS chat, which is all those in the COMMON file. Then test all the binds in the Alt bindfile, using the Reference list or a list of notes to remind you.

- Faults are almost always one of two things: the game can't find a file; or, a newly defined bind has an error in it. Look for spaces in filenames and correct syntax in bind definitions.
- Again, you can switch over to the text editor and make bindfile corrections and changes, then save it and reload as above. If you need to reload any but the Alt bindfile, follow the sequence above under (2). The Alt bindfile can be reloaded as many times as necessary using the **SHIFT+F3** key.
- After each successful test, save the composite bindfile with **Alt-F5**. That gives you a quick reload option, and a "filtered" version of the binds for review and troubleshooting. These files are saved under the name **[ALTNAME]-load.txt**.

## Keybind Extension & Maintenance

Once you the common and basic alt binds working smoothly, it's time to start extending the set until it covers all of your alt's powers and capabilities, and in a way that makes those powers a breeze to manage even in the heat of combat. There's no one path for this, but here are some good starting points:

- Simply assigning frequently used powers to a few standardized keys can streamline activation. I often assign buffs and power-ups to the **U**, **I**, **O** and **P** keys, for example, and it's easy to be consistent by general type across alts and powersets, making it less of a chore to remember what key does what.
- **Rollover Binds** like the one-key Rest control on **EQUALS** is a good example of how a need can be simplified, streamlined and add useful aspects such as automatically notifying teammates when an alt goes in and out of Rest mode.
- More extended **Rolling Binds** like those used for the chat strings can be applied in many ways, to activate several powers in a row or vary attacks among three or four primary strikes.
- **Dual-Action Binds** can be used to do things like fire two or three combat buffs (**Aim**, **Charge Up**, **Build Up**, etc.) and then fire a primary attack. My blaster, for example, can do a first-strike attack with pressing **ALT-P** (fire **Aim**), release **ALT-P** (fire **Build Up**), then pressing **P** (fire **Zapp**). *Tick-tick-tick-BAM!* with very little need to think the actions through or ways to make a mistake.

- **Mouse-Key Binds** can put travel, attack and support powers right under your fingertips. If there's a better way to teleport than point and double click, I don't know what it is.
- And finally, there are **Walking Binds**, that when combined with your forward motion key **W**, mean no buff, shield, defense or power ever needs to drop or go unactivated again. Simply moving forward in a natural manner keeps as many toggles and click-powers activated as needed. This is useful on almost every build, and priceless on toggle-heavy tanks and defenders.

Besides looking at sample bind files and discussion in the HC forums, the Tech Guide contains many sample binds and details of how to use these advanced techniques to get the most efficiency and QoL from your time in virtual spandex.

## This Alt is Bound for Glory: Down the Road

As your alt gains new powers, it's good to plan and find useful ways to bind them to available keys. And with this new two-part bindfile system, it's easy to add selected commands to all your alts, without having to edit every one's bind file, and keep even the most elaborate, specialized binds exclusive to one alt.

Once you have this system implemented across at least your major alts, it's good practice to do the clear-and-load process each time you dust one off. That way, any tweaks you put in the common file will be loaded, and any tweaks you might have made to the alt binds will be ready to go.

Not sure how to do something, or if it can be done? Just ask in the HC forums, and you'll find out. Can't hurt to flag the post with my username (**@Shenanigunner**) to make sure I take a look!

## The GABS Bindfile Reference List

The complete listing of both the COMMON and base ALTNAME bindfiles begins on the next page. Although a bit complicated, it's done with paired lines all the way through so that you can see what's in each file and start to understand the why of it all. Sorry this required an awkward page size, but it was that or squeeze things to unreadability!

	COMMON BINDFILE	ALT BINDFILE
<b>Files begins with a comment, which is any line that begins with a semicolon.</b> While a semicolon is a valid binding key, it is not used in these examples and makes a simple, one-character choice. Each comment line will be read in and assigned to <code>;</code> so a final statement to null the assignment, or write a desired keybind to it, is necessary.	1 ; GABS COMMON BIND LOAD FILE v1.0.0 2 3 4 5	1 ; GABS [ALTNAME] LOAD FILE v1.0.0 2 3 4 5
Unused keys disabled with <b>nop</b> statements. These keys may be assigned with or without shift combinations but are generally not used for binds.	6 7 ' "nop" 8 - "nop" 9 ALT+- "nop" 10 \ "nop" 11 COMMA "nop" 12 INSERT "nop" 13 PERIOD "nop" 14 SCROLL "nop" 15 PAUSE "nop"	6 7 8 9 10 11 12 13 14 15
Start chat with an opening slash for slash command entry.	16 17 / "show chat\$\$slashchat" 18	16 17 18
Quick-enter "Yes/OK" to dialogs. Quick enter "No/Cancel" to dialogs.	19 [ "dialog_no\$\$dialog_answer Decline\$\$dialog_answer Ok" 20 ] "dialog_yes\$\$dialog_answer Accept" 21	19 20 21
The base entry code can be set to your own SG's, or one of the transport bases like this one. (That's the apostrophe, above <b>TAB</b> , by the way.)	22 23 24	22 ` "enterbasefrompasscode COSMICTRANSPORT-6608" 23 24
The <b>1</b> to <b>0</b> keys (along the keyboard top) are configured to fire the power in the associated primary tray, an with <b>CTRL</b> , the power in the secondary tray. The <b>ALT</b> and <b>SHIFT</b> options for each are nop'ed out. Note that the key numbering is <b>1</b> to <b>0</b> while the slot numbers are <b>1</b> to <b>10</b> . There are two numbering systems in binds, zero-based and one-based. It's important to keep them straight. The game exports <b>0</b> first, then 1, and in the higher numbers, then <b>10</b> , <b>11</b> and <b>12</b> . It can get confusing but this bind file has been reordered numerically. (The load order does not matter, but if you export a fully loaded bindfile, the game will rewrite the order... just be aware of that.)	25 1 "powexec_slot 1" 26 CTRL+1 "powexec_altslot 1" 27 28 2 "powexec_slot 2" 29 CTRL+2 "powexec_altslot 2" 30 31 32 3 "powexec_slot 3" 33 CTRL+3 "powexec_altslot 3" 34 35 36 4 "powexec_slot 4" 37 CTRL+4 "powexec_altslot 4" 38 39 40 41 5 "powexec_slot 5" 42 CTRL+5 "powexec_altslot 5" 43 44 45	25 26 27 ALT+1 "nop" 28 SHIFT+1 "nop" 29 30 31 ALT+2 "nop" 32 SHIFT+2 "nop" 33 34 35 ALT+3 "nop" 36 SHIFT+3 "nop" 37 38 39 ALT+4 "nop" 40 SHIFT+4 "nop" 41 42 43 ALT+5 "nop" 44 SHIFT+5 "nop" 45
This is a good place to talk about <b>nop</b> vs <b>null</b> . The <b>nop</b> keyword, seen in the examples at right, is a specific keyword that means "do nothing." It should be used—and is used here—when a specific key should be rendered completely "dead."		
The alternative is to leave a key undefined, or <i>null</i> , by giving it an empty string (""), which we'll see in commands on the next pages. The result is the same—the key will do nothing—but in a large and complex bindfile that might push the overall character and line limit, it saves space not to have dozens of three-letter keywords padding things out.		

The one place a **nop** *must* be used instead of a null is for any combination where a primary key is assigned but the **SHIFT**ed key is not. If, say, **6** is assigned but **SHIFT+6** is left null, both combinations will fire the command assigned to 6, which can be unintended. Using **nop** renders the combination key 'dead.' (This does not apply to **ALT** or **CTRL**.)

	COMMON BINDFILE	ALT BINDFILE
	1 6 "powexec_slot 6"	1
	2 CTRL+6 "powexec_altslot 6"	2
	3	3 ALT+6 "nop"
	4	4 SHIFT+6 "nop"
	5 7 "powexec_slot 7"	5
	6 CTRL+7 "powexec_altslot 7"	6
	7	7 ALT+7 "nop"
	8	8 SHIFT+7 "nop"
	9 8 "powexec_slot 8"	9
	10 CTRL+8 "powexec_altslot 8"	10
	11	11 ALT+8 "nop"
	12 9 "powexec_slot 9"	12 SHIFT+8 "nop"
	13 CTRL+9 "powexec_slot 9"	13
	14	14
	15	15 ALT+9 "nop"
Again, in export order, the <b>0</b> set will come before the <b>1-9</b> set.	16 0 "powexec_slot 10"	16 SHIFT+9 "nop"
	17 CTRL+0 "powexec_altslot 10"	17
	18	18 ALT+0 "nop"
	19	19 SHIFT+0 "nop"
	20	20
Motion binds have been universally simplified. <b>A</b> is strafe left; <b>CTRL+A</b> is actual "turn alt's viewpoint to the left." A useful distinction, sometimes. See <b>D</b> , <b>S</b> and <b>W</b> . The other combo binds for movement keys should not be used.	21 A "+left"	21
	22 CTRL+A "+turn_left"	22
	23 ALT+A "nop"	23
	24 SHIFT+A "nop"	24
<b>Gunner's Targeting Bind.</b> <b>CTRL+ADD</b> (the numpad <b>+</b> key) will start a chat for you to type the first few characters of a target foe name. Hit <b>[Return]</b> and <b>ADD</b> will now search/target that foe type. The base version, copied on <b>ALT+ADD</b> , will search for objects like glowies and doors.	25	25 ADD "target_custom_next base"
	26	26 CTRL+ADD "beginchat "/bind add target_custom_next enemy alive "
	27	27 SHIFT+ADD "nop"
	28	28 ALT+ADD "target_custom_next base"
	29	29
	30	30 B "nop"
All other <b>B</b> key combos usable for alt binds.	31 CTRL+B ""	31
	32	32 SHIFT+B ""
<b>ChatBinds</b> — the battlecry bind is moved here. You can <i>edit</i> the parameters, and use a battlecry string instead of the defined variable.	33 ALT+B "1 &lt;color #00ccff&gt;&lt;bgcolor #0000000&gt;&lt;scale 4&gt;&lt;duration 5&gt;\$battlecry"	33 ALT+B "1 &lt;color #00ccff&gt;&lt;bgcolor #0000000&gt;&lt;scale 4&gt;&lt;duration 5&gt;\$battlecry"
	34	34
	35	35
Reply to last tell <i>received</i> .	36 BACKSPACE "autoreply"	36
Reply to last tell sent.	37 CTRL+BACKSPACE "tell_last"	37
Initiate tell to targeted player.	38 ALT+BACKSPACE "t \$target, "	38
	39	39 SHIFT+BACKSPACE ""
	40	40
	41	41
If your mouse has additional buttons, they are very useful;	42 BUTTON4 "powexec_toggleon Sprint\$\$autorun 1"	42
I find starting and stopping a travel power to be ideal.	43 BUTTON5 "powexec_toggleoff Sprint\$\$autorun 0"	43
Make the "go" button the harder one to activate.	44	44
	45	45

	COMMON BINDFILE	ALT BINDFILE
All other <b>C</b> key combos usable for alt binds but <b>do not use Ctrl-C</b> .	1 2 CTRL-C "nop" 3 4 5	1 C "nop" 2 ; DO NOT BIND CTRL-C 3 ALT+C "" 4 SHIFT+C "" 5
Motion binds have been universally simplified. <b>D</b> is strafe right; <b>CTRL+D</b> is actual "turn alt's viewpoint to the right." A useful distinction, sometimes. See <b>D</b> , <b>S</b> and <b>W</b> . The other combo binds for movement keys should not be used.	6 D "+right" 7 CTRL+D "+turn_right" 8 ALT+D "nop" 9 SHIFT+D "nop" 10	6 7 8 9 10
Decimal point on numpad. May be used with shift keys as well.	11 12	11 DECIMAL "nop" 12
Clear all chat windows.	13 DELETE "clearchat" 14	13 14
Divide slash on numpad. May be used with shift keys as well.	15 16	15 DIVIDE "nop" 16
A quirk of the bindfile key names makes it wise to bind both versions of this key to the same command string; see also <b>LEFT</b> , <b>RIGHT</b> , <b>UP</b> . Do not bind shifted versions unless you map these keys to other functions.	17 DOWN "+backward" 18 DOWNARROW "+backward" 19 20	17 18 19 20
A parallel key to <b>W</b> (forward) that allows forward motion without triggering walk-binds. See <b>W</b> for more information. It and all other <b>E</b> keys usable for alt binds.	21 E "+forward" 22 23 24	21 22 CTRL+E "" 23 ALT+E "" 24 SHIFT+E ""
Set viewpoint camera at maximum zoom. See <b>HOME</b> .	25 END "camdist 120" 26	25 26
Start chat in current chat window, pop custom AFK bubble. Bubble text may be <b>edited</b> to suit your preference and alt.	27 ENTER "show chat\$\$afk Speaking, excuse me...\$\$startchat" 28 29	27 ENTER "show chat\$\$afk Speaking, excuse me...\$\$startchat" 28 29
<b>EQUALS</b> is used to toggle <b>Rest</b> , configurable in each alt's binds. Complex and uses a rolling bind; see " <b>Rest Setup</b> " for details. Other <b>EQUALS</b> combo keys can be used for alt binds but should be avoided.	30 31 32 33 34 35	30 EQUALS "powexectoggleon Rest\$\$sg &lt;color white&gt;&lt;bgcolor red&gt;&lt;scale 2&gt;&lt;bordercolor white&gt;- RESTING! 32 -\$bindloadfilesilent [ALTNAME]-REST-OFF.txt" 33 CTRL+EQUALS "" 34 ALT+EQUALS "" 35 SHIFT+EQUALS ""
General cancel and "unselect selected thing" key; recommend not changing. See also <b>Z</b> , which aborts queued powers.	36 ESC "unselect" 37 38	36 37 38
Follow targeted object. See also <b>G</b> .	39 F "follow" 40	39 40 CTRL+F ""
<b>ChatBinds</b> — alert teammates the Boss has entered the room. Other <b>F</b> key combos may be used for alt binds.	41 ALT+F "g ### SPOTTED: \$target ###\$\$em whistle" 42 43 44 45	41 42 SHIFT+F "" 43 44 45

	COMMON BINDFILE	ALT BINDFILE
Fire a green (HEALTH) Inspiration, from lowest on hand to highest (of first three tiers; these commands can be extended). Load and save options specific to this alt.	1 F1 "inspexec_name resurgence\$\$inspexec_name dramatic	1
	2 improvement\$\$inspexec_name respite"	2
	3	3 CTRL+F1 "optionloadfile [ALTNAME]-options.txt"
	4	4 ALT+F1 "optionsavefile [ALTNAME]-options.txt"
General bindfile loader. Type in filename after calling up.	5 SHIFT+F1 "beginchat /bindloadfile .txt"	5
Fire a blue (ENDURANCE) Inspiration (Tier 1 to Tier 3). Other <b>F2</b> key combos usable for alt binds.	6	6
	7 F2 "inspexec_name second wind\$\$inspexec_name take a breather\$\$inspexec_	7
	8 name catch a breath"	8
	9	9 CTRL+F2 ""
	10	10 ALT+F2 ""
Load this common, base bindfile. (Follow with alt-file load.)	11 SHIFT+F2 "sg COMMON bindfiles loaded.\$\$bindloadfilesilent COMMON.txt"	11
Fire a lavender (BREAK FREE) Inspiration (Tier 1 to Tier 3). Other <b>F3</b> key combos usable for alt binds.	12	12
	13 F3 "inspexec_name escape\$\$inspexec_name emerge\$\$inspexec_name break free"	13
	14	14 CTRL+F3 ""
	15	15 ALT+F3 ""
Load this alt-specific, base bindfile when edits have been made.	16 SHIFT+F3 "nop"	16 SHIFT+F3 "sg ALT FILE loaded.\$\$bindloadfilesilent [ALTNAME].txt"
Fire a red (DAMAGE) Inspiration (Tier 1 to Tier 3). Other <b>F4</b> key combos usable for alt binds.	17	17
	18 F4 "inspexec_name righteous rage\$\$inspexec_name focused rage\$\$inspexec_	18
	19 name enrage"	19
	20	20 CTRL+F4 ""
	21	21 SHIFT+F4 ""
	22	22 ALT+F4 ""
	23	23
Fire a yellow (AIM) Inspiration (Tier 1 to Tier 3). Other <b>F5</b> key combos usable for alt binds.	24 F5 "inspexec_name uncanny insight\$\$inspexec_name keen insight\$\$inspexec_	24
	25 name insight"	25
Save this alt's composite, complete bindfile.	26	26 CTRL+F5 ""
	27	27 ALT+F5 "bindsavefile [ALTNAME]-load.txt"
	28	28 SHIFT+F5 ""
Fire an orange (DEFENSE) Inspiration (Tier 1 to Tier 3). Other <b>F6</b> key combos usable for alt binds.	29 F6 "inspexec_name robust\$\$inspexec_name rugged\$\$inspexec_name sturdy"	29
	30	30
Load this alt's composite, complete bindfile (after save).	31	31 CTRL+F6 ""
	32	32 ALT+F6 "bindloadfilesilent [ALTNAME]-load.txt"
	33	33 SHIFT+F6 ""
Fire a purple (NO-HIT) Inspiration (Tier 1 to Tier 3). Other <b>F7</b> key combos usable for alt binds.	34 F7 "inspexec_name phenomenal luck\$\$inspexec_name good luck\$\$inspexec_name	34
	35 luck"	35
	36	36 CTRL+F7 ""
	37	37 SHIFT+F7 ""
	38	38 ALT+F7 ""
	39	39
Toggle open/closed <b>Auction House</b> and your	40 F8 "ah\$\$toggle salvage\$\$toggle recipes"	40
<b>Recipe</b> and <b>Salvage</b> windows.	41	41 CTRL+F8 ""
Toggle open/closed a basic monitor window. The contents of	42	42 ALT+F8 "monitorattribute Endurance Consumption\$\$monitorattribute Recovery
this window can be edited, rearranged and added to.	43	43 Rate\$\$monitorattribute Regeneration Rate\$\$monitorattribute Running
	44	44 Speed\$\$monitorattribute Flying Speed\$\$monitorattribute Jump Height"
	45	45 SHIFT+F8 ""
Other <b>F8</b> key combos usable for alt binds.		

	COMMON BINDFILE	ALT BINDFILE
	1	1
Exit mission (equivalent to clicking teeny-tiny blue button in Nav window). Other <b>F9</b> key combos usable for alt binds.	2 <b>F9</b> "requestexitmission 1"	2
	3	3 CTRL+F9 ""
	4	4 SHIFT+F9 ""
Bring up game exit menu.	5 <b>ALT+F9</b> "show quit"	5
	6	6
Set basic viewing distance and filtering. Values may be edited to suit. Other <b>F10</b> key combos usable for alt binds.	7	7 <b>F10</b> "visscale 5\$\$dofweight 0.5\$\$fsaa 4"
	8	8 CTRL+F10 ""
	9	9 ALT+F10 ""
	10	10 SHIFT+F10 ""
	11	11
All <b>F11</b> key combos usable for alt binds.	12	12 <b>F11</b> "nop"
	13	13 CTRL+F11 ""
	14	14 ALT+F11 ""
	15	15 SHIFT+F11 ""
	16	16
Load this specific alt's window/UI setup (after save). Make map window large for navigation (adjust value to suit). Save current windows/UI arrangement for this specific alt. Set window border color and opacity (RGBT).	17	17 <b>F12</b> "wdw_load_file [ALTNAME]-windows.txt"
	18	18 CTRL+F12 "window_scale map 2"
	19	19 ALT+F12 "wdw_save_file [ALTNAME]-windows.txt"
	20	20 SHIFT+F12 "window_color 0 140 150 150"
	21	21
<b>THE</b> melee bind - "Grapple." Locks you onto the nearest foe in combat. Other <b>G</b> key combos usable for alt binds.	22 <b>G</b> "target_enemy_near\$\$follow"	22
<b>ChatBinds</b> — a rolling (changing) "Gratz!" message.	23	23 CTRL+G ""
	24 <b>ALT+G</b> "g Gratz!\$\$bindloadfilesilent rollchat2.txt"	24
	25	25 SHIFT+G ""
	26	26
Suggested for binding to primary self-heal or -buff. <b>NOP</b> out otherwise. Other <b>H</b> key combos usable for alt binds.	27 <b>H</b> "nop"	27 ; H "powexecname Healing Flames"
<b>ChatBinds</b> — a rolling (changing) "Held!" message.	28	28 CTRL+H ""
	29 <b>ALT+H</b> "g !!! HELD !!!\$em whistle\$\$bindloadfilesilent rollchat2.txt"	29
	30	30 SHIFT+H ""
	31	31
Set viewpoint camera at a comfortable distance. Value may be edited. See <b>END</b> .	32 <b>HOME</b> "camdist 20"	32 HOME "camdist 20"
	33	33
All <b>I</b> key combos usable for alt binds.	34	34 <b>I</b> "nop"
	35	35 CTRL+I ""
	36	36 ALT+I ""
	37	37 SHIFT+I ""
	38	38
All <b>J</b> key combos usable for alt binds.	39	39 <b>J</b> "nop"
	40	40 CTRL+J ""
	41	41 ALT+J ""
	42	42 SHIFT+J ""
	43	43
	44	44
	45	45

	COMMON BINDFILE	ALT BINDFILE
All <b>K</b> key combos usable for alt binds.	1 2 3 4 5	1 K "nop" 2 CTRL+K "" 3 ALT+K "" 4 SHIFT+K "" 5
All <b>L</b> keys usable for alt binds.	6 7 8 9 10	6 L "nop" 7 CTRL+L "" 8 ALT+L "" 9 SHIFT+L "" 10
Don't use left and right control keys as tap binds. They're buggy. Note another instance of duplicated keybind names. Bind both the same.	11 LALT "nop" 12 LCONTROL "nop" 13 LCTRL "nop" 14	11 12 13 14
A quirk of the bindfile key names makes it wise to bind both versions of this key to the same command string; see also <b>DOWN, RIGHT, UP</b> .	15 LEFT "+left" 16 LEFTARROW "+left" 17	15 16 17
The mouse button binds (including BUTTON4/5 above) are complex. These examples from an alt with the Teleport power set have been left in place as examples; they should be edited to reflect actual travel powers, or nulled out to prevent faulty commands.	18 LeftDoubleClick "nop" 19 CTRL+LeftDoubleClick "nop" 20 ALT+LeftDoubleClick "nop" 21 22	18 ; LeftDoubleClick "powexec_location cursor Teleport" 19 ; CTRL+LeftDoubleClick "powexec_location cursor Combat Teleport" 20 ; ALT+LeftDoubleClick "powexecname Teleport Target\$\$g Teleporting \$target to me!" 21 22 SHIFT+LeftDoubleClick ""
Turn off travel power and toggle basic run power (Sprint or equivalent).	23 24	23 MBUTTON "powexectoggleoff Athletic Run\$\$powexecname Sprint" 24 RightDoubleClick ""
See <a href="#">Mouse Buttons</a> for further details.	25 26 27	25 CTRL+RightDoubleClick "" 26 ALT+RightDoubleClick "" 27 SHIFT+RightDoubleClick ""
Jump or fly upwards with a click of both mouse buttons. Adjust viewpoint zoom and turn player to face pointed direction. Scroll through enemies, forward... ...and backward.	28 MouseChord "+up" 29 MOUSEWHEEL "+camdistadjust\$\$playerturn" 30 CTRL+MOUSEWHEEL "target_enemy_next" 31 SHIFT+MOUSEWHEEL "target_enemy_prev" 32	28 29 30 31 32
Toggle <b>Contact</b> and Mission windows open/closed. All other <b>M</b> key combos usable for alt binds.	33 M "toggle contact\$\$toggle mission" 34 35	33 34 CTRL+M "" 35 ALT+M ""
Costume change shortcuts. Assign desired costume slot and emote to each.	36 37 38 39 40 41 42 43 44 45	36 SHIFT+M "" 37 38 MULTIPLY "cce 1 ccNuke" 39 CTRL+MULTIPLY "cce 2 ccSmokeBomb" 40 SHIFT+MULTIPLY "cce 3 ccInnerWill" 41 ALT+MULTIPLY "cce 5 ccRainbow" 42 43 44 45

	COMMON BINDFILE	ALT BINDFILE
All other <b>N</b> key combos usable for alt binds.	1	1 N "nop"
Show and hide your alt's name display.	2 CTRL+N "option_toggle showownername"	2
<b>ChatBinds</b> — a rolling (changing) "No" message.	3 ALT+N "l No, thx.\$bindloadfilesilent rollchat2.txt"	3
	4	4 SHIFT+N ""
	5	5
All numpad keys are available for alt binds, including all three shift key alternates for each. The keypad grid is excellent for team heals, buffs and pet control	6	6 NUMPAD0 ""
	7	7 NUMPAD1 ""
	8	8 NUMPAD2 ""
	9	9 NUMPAD3 ""
	10	10 NUMPAD4 ""
	11	11 NUMPAD5 ""
	12	12 NUMPAD6 ""
	13	13 NUMPAD7 ""
	14	14 NUMPAD8 ""
	15	15 NUMPAD9 ""
	16	16 NUMPADENTER ""
	17	17
All <b>O</b> key combos usable for alt binds.	18	18 O "nop"
	19	19 CTRL+O ""
	20	20 ALT+O ""
	21	21 SHIFT+O ""
	22	22
All <b>P</b> key combos usable for alt binds.	23	23 P "nop"
	24	24 CTRL+P ""
	25	25 ALT+P ""
	26	26 SHIFT+P ""
	27	27
Turn camera to face direction alt is looking.	28 PAGEDOWN "cam_turn"	28
Hide the <b>Team</b> window.	29 CTRL+PAGEDOWN "windowhide team"	29
Turn alt to face direction viewpoint/camera is looking.	30 PAGEUP "playerturn"	30
Show the <b>Team</b> window. Scale factor may be edited to suit your preference.	31 CTRL+PAGEUP "show team\$\$windowsscale team 1"	31 CTRL+PAGEUP "show team\$\$windowsscale team 1"
	32	32
All <b>Q</b> key combos usable for alt binds.	33	33 Q "nop"
	34	34 CTRL+Q ""
	35	35 ALT+Q ""
	36	36 SHIFT+Q ""
	37	37
Toggle on run power and, well, run.	38	38 R "powexectoggleon Sprint\$\$autorun 1"
	39	39 CTRL+R ""
<b>ChatBinds</b> — a rolling (changing) "Ready!" message.	40 ALT+R "g Ready!\$\$em knuckles\$\$bindloadfilesilent rollchat3.txt"	40
All other <b>R</b> key combos usable for alt binds.	41	41 SHIFT+R ""
	42	42
	43	43
	44	44
	45	45

	COMMON BINDFILE	ALT BINDFILE
Again, don't use the control keys as tap binds. Buggy.	1 RALT "nop"	1
	2 RCONTROL "nop"	2
	3 RCTRL "nop"	3
	4	4
Default hold right mouse button to look around. Should not be changed.	5 RBUTTON "+mouse_look"	5
	6	6 CTRL+RBUTTON ""
Unlock camera to pivot view around your alt.	7 SHIFT+RBUTTON "+camrotate"	7
Binding other <b>RBUTTON</b> combos not recommended.	8	8 ALT+RBUTTON ""
	9	9
A quirk of the bindfile key names makes it wise to bind both versions of this key to the same command string; see also <b>DOWN</b> , <b>LEFT</b> , <b>UP</b> .	10 RIGHT "+right"	10
	11 RIGHTARROW "+right"	11
	12	12
React properly to a boss that cons purple: retreat! See <b>A</b> , <b>D</b> and <b>W</b> .	13 S "+backward"	13
The other combo binds for movement keys should not be used.	14 CTRL+S "nop"	14
	15 ALT+S "nop"	15
	16 SHIFT+s "nop"	16
	17	17
Jump, and cancel autorun.	18 SPACE "+up\$\$autorun 0"	18
	19	19
Minus key on keypad. Like <b>ADD</b> , great for quick right-hand thumb binds like search.	20	20 SUBTRACT "nop"
All <b>SUBTRACT</b> key combos may be used for alt binds.	21	21 CTRL+SUBTRACT ""
	22	22 ALT+SUBTRACT ""
	23	23 SHIFT+SUBTRACT ""
	24	24
Take screenshot (will appear in <b>[GAME FOLDER]\screenshots</b> ).	25 SYSRQ "screenshot"	25
Set screenshots to fire with UI toggled off.	26 CTRL+SYSRQ "screenshotui 0"	26
	27	27
Target nearest enemy.	28 T "target_enemy_near"	28
Target next enemy.	29 CTRL+T "target_enemy_next"	29
<b>ChatBinds</b> — a rolling (changing) "Thanks!" message.	30 ALT+T "1 Thanks!\$\$bindloadfilesilent c:\games\coh\bindfiles\rollchat2.txt"	30
	31	31
All other <b>T</b> key combos usable for alt binds.	32	32 SHIFT+T ""
	33	33
Toggle UI on and off.	34 TAB "++disable2D"	34
All other <b>TAB</b> key combos usable for alt binds.	35	35 CTRL+TAB ""
	36	36 ALT+TAB ""
	37	37 SHIFT+TAB ""
	38	38
	39	39 U "nop"
Turn off all activated powers and cancel any queued power.	40 CTRL+U "powerstogglealloff\$\$powexec_abort"	40
All other <b>U</b> key combos usable for alt binds.	41	41 ALT+U ""
	42	42 SHIFT+U ""
	43	43
A quirk of the bindfile key names makes it wise to bind both versions of this key to the same command string; see also <b>DOWN</b> , <b>LEFT</b> , <b>RIGHT</b> .	44 UP "+forward"	44
	45 UPARROW "+forward"	45

	COMMON BINDFILE	ALT BINDFILE
	1	1
Toggle 'Click to Move' (click on go-to spot) on and off.	2 V "+ctm"	2
All other <b>V</b> key combos usable for alt binds but <b>do not use Ctrl-V</b> .	3 CTRL-V "nop"	3 ; DO NOT BIND CTRL-V!
	4	4 ALT+V ""
	5	5 SHIFT+V ""
	6	6
	7	7
	8	8
Forward — basic or very advanced. See <b>Walking Binds</b> .	9 W "+forward"	9 W "+forward"
	10	10 ; W "+forward\$\$powexectoggleon Fire Shield\$\$bindloadfilesilent [ALTNAME]-
	11	11 WALK-B3.txt"
	12	12 CTRL+W ""
<b>ChatBinds</b> — a rolling (changing) "Wait up!" message.	13 ALT+W "g Hold up a bit!\$\$em whistle\$\$bindloadfilesilent rollchat2.txt"	13
All other <b>W</b> key combos usable for alt binds.	14	14 SHIFT+W ""
	15	15
Down. Only useful for hover/flight.	16 X "+down"	16
All other <b>X</b> key combos usable for alt binds but <b>do not use Ctrl-X</b> .	17 CTRL+X "nop"	17 ; DO NOT BIND CTRL-X!
	18	18 ALT+X ""
	19	19 SHIFT+X ""
	20	20
Zoom key, adapt to specific travel power. See <b>Zoom!</b> .	21 Y "nop"	21 ; Y "powexec_location up:max Teleport"
	22	22 CTRL+Y ""
<b>ChatBinds</b> — a rolling (changing) "Yes" message.	23 ALT+Y "l Yes\$\$bindloadfilesilent rollchat2.txt"	23
All other <b>Y</b> key combos usable for alt binds .	24	24 SHIFT+Y ""
	25	25
Cancel any queued power and cancel auto-fire power.	26 Z "powexec_abort"	26
All other <b>Z</b> key combos usable for alt binds .	27	27 CTRL+Z ""
	28	28 ALT+Z ""
	29	29 SHIFT+Z ""
	30	30
<b>As noted at the start</b> , if semicolon lines are used for comments, the very last line of the bindfile must be this <b>NOP</b> or a valid key assignment.	31 ; "nop"	31 ; "nop"
	32	32

## Bindfile Basics

It is extremely important that all bindfiles — primary, secondary, 'rolling' or whatever — be (1) plain ASCII text, with (2) plain single and double quotes, not "curly" ones, and (3) placed in **[GAME FOLDER]\settings\live** so that bind and bindfile calls can find them.

Comments may be included in bindfiles, on single lines (with no returns or hard wrap) beginning with a semicolon (;). As this is a valid binding key name, the **very last line in any bindfile that uses comments must be—**

```
; "nop"
```

—or a valid keybind to the semicolon key.

If anything in this guide, the bindfiles or having to do with the character UI doesn't make sense, the answer is probably in the **Tech Guide**, which you should have handy as an advanced player anyway.

## NOP vs <null>

Binds can be defined to do nothing by using either a **nop** or a **<null>**.

The **nop** keyword is a specific command that means "do nothing" ("no operation"). It should be used—and is used here—when a specific key should be rendered completely "dead." It will not be overridden by the mapping of other keys, and it is persistent in that the 'nopped' definition will be preserved when the keybind set is saved or exported.

The alternative is to leave a key undefined, or *null*, by giving it an empty string (""). The result is the same—the key will do nothing, but only most of the time.

There is at least one oddity with a null definition: if a base key is assigned a function, and SHIFT+ that base key is left null, then both combinations will actuate the base function. (That is, both B and SHIFT-B would activate the bind assigned to B.) This is trivial, but in a complex layout, may have unintended results. **For critical functions, the SHIFT combination should be nopped, not nulled.** (And for some reason, this is unique to SHIFT and does not apply to CTRL or ALT.)

Using the null string has two other desirable effects, though. The first is saving quite a few characters, three at a time, and in very large bindfiles pushing the character and line limit, that's not trivial. The second is that

when a bindfile is saved or exported, nulled entries simply disappear—a definition like

```
CTRL+Insert ""
```

will simply not appear in the exported bindfile. (Technically, it's never loaded in the first place, which can also help bind files stay under the size limits.) This can clean out a lot of "dead" lines that will probably never be used for that bindfile.

Pay attention to which commands are given **nops** and which are *nulled* in the sample files. The difference, and reasons, should become clear.

## Rolling ChatBinds

This bindfile uses a half-dozen keys for common chat responses — 'Yes,' 'No,' 'Gratz!' etc. To keep the use of these buttons from becoming tedious (or even obvious to others that they're canned) three exterior files are used to "roll" the binds. Pressing **ALT-Y** three times will thus send 'Yes,' 'Yep' and 'Affirmative' before repeating. The keys are mnemonically assigned to keep the system easy to use.

The function can be understood by looking at any of the three rolling files — **rollchat1.txt**, **rollchat2.txt** and **rollchat3.txt** — to see what the text strings are and how the next rolling file is called.

This could be easily extended to four, five or ten responses if you want to create and carefully link the files. You can always duplicate some phrases if you can't think of ten ways to say "Wait up!"

**Note that any use of a chat bind will roll all of them to the next file. It would be insanely complicated to give each phrase its own set of files.**

One dilemma you will have to resolve is which channel in which to speak your messages. If you use Team (**g**), you won't be able to say Yes, No or Thanks to anyone outside the team. If you use Local (**l**), you may look like a babbling fool to others outside of your team. But if you use Team and are soloing... each message will just generate a "You are not on a team!" error message, which is annoying.

My solution, which is imperfect, is to put the common messages (Yes, No, Thanks!, Gratz!) in Local, where both teammates and other players can see them. Since the other messages (Held!, Ready!, Spotted, Wait Up) only make sense to a team, there they are on Team.

One clumsy workaround, useful if you mostly solo, as I do, is to assign the team messages to the SuperGroup (**sg**) channel, so they show up when soloing. The problem there is that anyone on a team who isn't an SG member won't see them, but other SG members will. I haven't found a good general solution. Let me know if you think of one.

## One-Command Rest Setup

This bindfile uses an elaborate system to call Rest; a single key toggles Rest on and off, with an appropriate message to teammates each time. Besides being quicker than most clicky-activations, having your Tank call out "RESTing my weary butt!" can keep teammates from charging into the next battle, not realising you aren't there to sop up aggro.

This is done with a pair of rolling bindfiles that can be simply constructed or made quite complicated. The simplest pair would be:

```
[MYALT-REST-ON.txt]
EQUALS "g RESTING!$$$powexectoggleon Rest$$bindloadfilesilent
MYALT-REST-OFF.txt"
```

```
[MYALT-REST-OFF.txt]
EQUALS "g LET'S GO!$$$powexectoggleoff Rest$$bindloadfilesilent
MYALT-REST-ON.txt"
```

A few minutes of studying those files should make it clear how they work. (If not, the **Tech Guide** has a whole section on rolling binds.)

As with the ChatBinds, choosing the right channel is a bit sticky. Team is the obvious choice; I use SuperGroup for the reasons mentioned above. You probably don't want to use Local.

Where this can get fancy is in styling the chat bubble and text with colors, size and duration. The code in the sample bindfiles (copied above in the listing) uses these, but it's all quite tricky. Rather than copy most of a Guide section here, I will just point you to **Reference E** in the Tech Guide.

Even there, though, there's one more sticky bit. You can't use angle brackets ( < > ) in bind strings, so you have to use the codes instead: **&lt;** for the opening bracket ("less than") and **&gt;** for the closing bracket ("greater than"). Look closely at the examples, and it should be clear(-ish).

But even the simple version is worth using!

## Walking Binds

### Welcome to the most hugely powerful Quality-of-Life enhancement you can apply to City of Spandex: walking power binds.

If you have any alt that has toggle powers (shields, buffs, etc.) and you’re tired of having to remember to turn them on, and wading into combat only to find something has dropped them, then Walking Binds are for you. Since there is hardly a build that doesn’t have toggles and click powers most players would like to keep active, maybe these are for everyone.

While it’s possible to bind, say, three shield powers to one key, they can only be activated one at a time. So it would take three presses of the P key are needed, and often not too quickly. Annoying and manual.

Some clever player, long ago, thought to bind a toggle power to the forward-walk button, so that every press to move forward would also re/activate that power. A clumsy solution that evolved was to bind one such power to each movement key, but to be honest, you can play a whole session without touching A or D and only tapping S once an hour. Since the whole point is that you shouldn’t have to remember to reactivate powers, a more “automatic” method is needed... so here it is.

```
/bind W +forward$$powextoggleon Focused
      Fighting$$bindloadfilesilent Alt-WALK-02.txt
```

```
[Alt-WALK-02.txt]
W +forward$$powextoggleon Focused Senses$$bindloadfilesilent
  Alt-WALK-03.txt
```

```
[Alt-WALK-03.txt]
W +forward$$powextoggleon Focused Senses$$bindloadfilesilent
  Alt-WALK-01.txt
```

```
[Alt-WALK-01.txt]
W +forward$$powexecauto Practiced Brawler$$bindloadfilesilent
  Alt-WALK-02.txt
```

See what’s going on, there? Every time the **W** button is pressed, three things happen:

- The alt moves forward until the key is released. (Normal.)
- A toggle power is turned on, or a click power is activated.

- A rollover bind replaces the **W** string so that on the next press, the process repeats but with a different power toggled on.

In other words, those three powers will never go off, or remain off for more than a minimum recovery time, as long as the player is moving the alt forward... which is probably 95% of actual playing time. The string of rollover files can be as long as needed—five, six, ten—to keep all of an alt’s toggle/click powers active. **Honestly, I think not using walking binds on pretty much every alt is just leaving Inf on the table. Lots of it.**

It is, of course, imperative to slot the click powers for as close to “perma” activation as possible, and if a slot can be spared, a recharge on any toggle powers with long recovery times is good, too.

Note that toggle powers use **powextoggleon**, while click powers use **powexecauto**. Technically, **...toggleon** can be used for both, but every time they get to a click power that is already activated, you’ll get that annoying “*whong*” sound from the console. Faking it on with a set to auto-fire will click it whenever it’s off, but the “*whong*” is bypassed.

There can be timing issues for certain powers and keyboard drivers. With just the rotation of activation strings, powers can be skipped repeatedly and other unwanted behavior can occur. So it helps to put a buffer file in between each activation file, like this:

```
[Alt-WALK-01.txt]
W +forward$$powexecauto Practiced Brawler$$bindloadfilesilent
  Alt-WALK-B1.txt
```

```
[Alt-WALK-B1.txt]
W +forward$$bindloadfilesilent Alt-WALK-02.txt
```

```
[Alt-WALK-02.txt]
W +forward$$powextoggleon Focused Senses$$bindloadfilesilent
  Alt-WALK-B2.txt
```

```
[Alt-WALK-B2.txt]
W +forward$$bindloadfilesilent Alt-WALK-03.txt
```

```
[Alt-WALK-03.txt]
W +forward$$powextoggleon Focused Senses$$bindloadfilesilent
  Alt-WALK-B3.txt
```

```
[Alt-WALK-B3.txt]
W +forward$$bindloadfilesilent Alt-WALK-01.txt
```

See? Each file with a power activation calls a null file that does not activate anything, and the null file calls the next activation file, round and round. These buffer files are all but undetectable in the actual working cycle, but keep the activated powers from bumping into one another.

Walking bindfiles must be individually named for each alt, unless you have two alts using the exact same powers. Even then, it’s much smarter to have a separate ring of files for each alt. The “Alt” in each filename and load call should be replaced with a short form of the Alt’s name.

**All powers can be turned off—to rest, to stop seeing mad particle fx, just to stand quietly—with a tap of the CTRL-U button. That’s what it’s for.**

## Dual-Action Binds

There’s another little-known trick to fire multiple powers from a single key, something that is deliberately not supported by the keybind/slash command structure of CoX. (The intent was to prevent all forms of automation, and it is largely successful. You can’t do anything purely on an automatic or timed loop; player input is required for each action to occur.) This trick relies on the game being able to detect the difference between a key *press* and a key *release* (which almost anyone who’s dabbled in programming knows are two different system-level events).

*(I originally called these press-release binds, until one too many smartasses asked where the press release was.)*

By simply putting a + sign at the beginning of a bind string, the command parser will execute the string once on key press, and again on key release. This can sometimes be used in a simple fashion, to get two identical actions for the price of one. That’s not very useful, but when combined with a rolling bind to switch the bind string in between press and release, two actions can be had with one keypress. Like this:

```
/bind ALT+P “+$$powexecname Charge Up$$bindloadfilesilent
  Glory-GPU-02.txt”
```

```
[Glory-GPU-02.txt]
ALT+P “+$$powexecname Build Up$$bindloadfilesilent
  Glory-GPU-01.txt”
```

```
[Glory-GPU-01.txt]
ALT+P “+$$powexecname Charge Up$$bindloadfilesilent
  Glory-GPU-02.txt”
```

The first press fires one buff (usually **Charge Up**, although the order can get reversed with use), and on release, fires **Build Up**. (As noted above, this combo is followed by a press of P to fire the blaster primary **Zapp**.)

Here’s another example from the same alt:

```
/bind 0 “+$$powexecname Power Sink$$bindloadfilesilent
      Glory-ZPU-02.txt”
```

```
[Glory-ZPU-02.txt]
0 “+$$powexecname Short Circuit$$bindloadfilesilent
  Glory-ZPU-01.txt”
```

```
[Glory-ZPU-01.txt]
0 “+$$powexecname Power Sink$$bindloadfilesilent Glory-ZPU-02.txt”
```

This pair fires the AoEs **Power Sink** and then **Short Circuit** to assault and power-drain all foes within range. Very effective.

One caveat is that too quick a keypress cycle may cause a failure to fire one or both powers. A deliberate press...pause...release is needed for reliable action. The pause need not be any measurable length, just deliberate enough to clearly separate the down and up. Not a tap.

## Mouse Buttons

The buttons on any mouse, 2-, 3- or multi-button, can be put to savvy use in controlling your alt’s actions. Combined with the relatively new **powexec\_location** statement, especially using the **cursor** keyword, the mouse and keys can be used for many travel and targeted actions.

The best and simplest might be “effortless teleportation”:

```
LeftDoubleClick “powexec_location cursor Teleport”
```

Point, double-click, go. Works for one-time movement or travel across a zone. A good companion bind is:

```
CTRL+LeftDoubleClick “powexec_location cursor Combat Teleport”
```

to pop right into contact with the next foe.

I am clumsy with middle and right double-clicking, but the commands are there if you’re not. I do use this bind:

```
RightDoubleClick “popmenu WindowScaler”
```

To open the **WindowScaler** menu to resize a UI window. It could be used for the various badge-hunter or emote popmenus as well. (WindowScaler is available on my website, along with the Tech Guide.)

It is best not to rebind any mouse key or key combo that’s standard in the game control—left click or drag, right click or drag.

While the mousewheel can be rebound, it tends to get cranky if not used for the base purpose of zooming in and out, so I’d leave it alone as well. (It could be useful bound, with a shift key, to **target\_enemy\_next** and **\_prev**, but I’ve found that to be more a ‘good’ idea than a useful one.

If you have the two extra buttons common to many modern mice, they can be put together in good service of travel power control:

```
BUTTON4 “powexectoggleon Super Speed$$autorun 1”
BUTTON5 “powexectoggleoff Athletic Run$$powexectoggleoff
        Super Speed$$autorun 0”
```

or

```
BUTTON4 “emote flypose1$$autorun 1$$powexectoggleon Mystic Flight”
BUTTON5 “autorun 0$$up 0$$powexectoggleoff Mystic Flight”
```

My practice is to bind the “go” power to the button that’s hardest to activate and takes a moment to locate, so that the “stop” power can be executed almost instantly. Especially with the flight powers, you can get very good at sticking the landing with this one-button-at-hand option.

Finally, one of the most useful bread-and-butter binds I know of is to use **MOUSECHORD**—the left and right mouse buttons pressed simultaneously—bound to **+up**. This puts jump and ‘rise in flight’ right under your fingers for effortless, one-handed travel and navigation.

## Zoom! Setup

A final bind worth noting is an instant-action one that can be adapted to many things, but which I’ve found ideal for a quick escape. When a battle gets out of control or you suddenly run into a player you’ve kicked from a team, leaving in the fastest way possible is a good idea.

I use the **Y** key for this but you’re free to reassign it. (I long used **Y** for fly

control, but I now put travel powers on the mouse buttons, as above.)

Zoom! requires a suitable travel power, with **Teleport** or **Translocation** being ideal, any flight power almost as good and **Super Jump** being acceptable. It could be adapted to the jet pack or steam jump as well, for alts that use running travel powers or no travel power option at all.

**Teleport** is simple:

```
Y “powexec_location up:max Teleport”
```

Tap **Y**, and you TP straight up to the limit of your power, or the nearest ceiling or overhang.

**Super Jump** is almost as easy:

```
Y “+up$$powexec_toggleon Super Jump”
```

The **up** command there is latched and will have to be canceled by tapping Space or any other key set to ‘up’ in a normal way. This same form can be used for **Fly** or the two jetpack powers.

If you have **Mystic Flight** or **Sonic Boom**, the **Translocation** power is only usable while the primary power is active. To enable this, you can use:

```
Y “powexec_location up:max Translocation$$powexectoggleon Mystic
  Flight”
```

If Mystic Flight is already on, the TP command will execute immediately. If flight is off, it will take two taps of **Y** to Zoom! out of trouble. And—*aha!*—you’re probably ahead of me on this:

```
Y “+$$powexec_location up:max Translocation$$powexectoggleon Mystic
  Flight”
```

Where a dual-action bind can start flight, if it’s off, and TP you up in a single keypress. If flight is already active, the TP will happen on the press, not the release, so just a blink quicker.

The first time you tap **Y** and are safe well above the mob of foes about to pound you to scrap, before you could even finish saying *Oh, Sh...*, you’ll really appreciate this bind.

Well... at least until one final ranged shot takes you out and you gracefully face-plant from 100 feet up. Hit **Y** faster next time.

# GABS v1.00 REFERENCE CHART

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
—	HEALTH	STAMINA	BREAK FREE	DAMAGE	AIM	DEFENSE BUFF	NO-HIT BUFF	AUCTION WINDOWS	EXIT MISSION	VIDEO SETTINGS		LOAD & RESET UI
C	OPTIONS LOAD											BIG MAP
A	OPTIONS SAVE				SAVE BINDS	LOAD BINDS		COMBAT MONITOR	QUIT MENU			SAVE UI
S	BF LOAD	BF LOAD COMMON	BF LOAD [ALTNAME]									WINDOW COLOR

	1	2	3	4	5	6	7	8	9	0	=
—	PRIMARY 1	PRIMARY 2	PRIMARY 3	PRIMARY 4	PRIMARY 5	PRIMARY 6	PRIMARY 7	PRIMARY 8	PRIMARY 9	PRIMARY 0	REST TOGGLE
C	SECONDARY 1	SECONDARY 2	SECONDARY 3	SECONDARY 4	SECONDARY 5	SECONDARY 6	SECONDARY 7	SECONDARY 8	SECONDARY 9	SECONDARY 10	

	B	E	F	G	H	M	N	R	T	U	W	Y
—		FORWARD (simple)	FOLLOW	GRAPPLE		MISSION/ CONTACTS		AUTORUN	TARGET NEAR	CANCEL POWERS	FORWARD	
C							TOGGLE NAME		TARGET NEXT			
A	<i>Battle Cry</i>		<i>"Found!"</i>	<i>"Gratz!"</i>	<i>"Held!"</i>		<i>"No"</i>	<i>"Ready!"</i>	<i>"Thanks!"</i>		<i>"Wait!"</i>	<i>"Yes!"</i>
S												

	BACKSPACE	ENTER	/	DEL	HOME	END	PGUP	PGDN	SYSRQ	TAB	ESC
—	REPLY TO TELL	CHAT	SLASH CHAT	CLEAR CHAT	SET CAMERA	ZOOM MAX	CAM TURN	ALT TURN	SCREEN SHOT	TOGGLE UI	CANCEL
C	FOLLOWUP TELL						SHOW TEAM	HIDE TEAM	HIDE UI IN S. SHOT		
A	TELL TARGET										
S											

	—	C	A	S
9				
8				
7				
6				
5				
4				
3				
2				
1				
0				
.				
ENTER				
+	MASTER TARGET			
-	NAME TARGET	NT DEFINE		
*	CCE 1	CCE 0	CCE 2	CCE 3
/				
[	Ok / No / Decline			
]	Yes / Accept			

A S D W & Arrow Keys: MOVE	CTRL-A/D: TURN	X: DOWN	SPACE: UP + AUTORUN OFF	RBUTTON: CAMLOOK	SHIFT+RBUTTON: FREE LOOK	MOUSECHORD: UP	MOUSEWHEEL: ZOOM
----------------------------	----------------	---------	-------------------------	------------------	--------------------------	----------------	------------------

ALT: \_\_\_\_\_

CTRL	<b>B</b>	
ALT		<i>Battle Cry</i> [local]
SHIFT		
ALT	<b>C</b>	
SHIFT		
		FORWARD
CTRL	<b>E</b>	
ALT		
SHIFT		
		FOLLOW
CTRL	<b>F</b>	
ALT		<i>"Found \$target!"</i> [team]
SHIFT		
		GRAPPLE
CTRL	<b>G</b>	
ALT		<i>"Gratz!"</i> [local]
SHIFT		
CTRL	<b>H</b>	
ALT		<i>"Held!"</i> [team]
SHIFT		
CTRL	<b>I</b>	
ALT		
SHIFT		
CTRL	<b>J</b>	
ALT		
SHIFT		
CTRL	<b>K</b>	
ALT		
SHIFT		
CTRL	<b>L</b>	
ALT		
SHIFT		

Simple movement

TargetNear+Follow

Primary self-heal

		Mission/Contact Toggle
CTRL	<b>M</b>	
ALT		
SHIFT		
CTRL	<b>N</b>	Show Alt Name
ALT		<i>"No."</i> [local]
SHIFT		
CTRL	<b>O</b>	
ALT		
SHIFT		
CTRL	<b>P</b>	
ALT		
SHIFT		
CTRL	<b>Q</b>	
ALT		
SHIFT		
CTRL	<b>U</b>	
ALT		
SHIFT		
ALT	<b>V</b>	Toggle CTM
SHIFT		
		DOWN
ALT	<b>X</b>	
SHIFT		
CTRL	<b>Y</b>	
ALT		<i>"Yes."</i> [local]
SHIFT		
		Abort Queued Powers
CTRL	<b>Z</b>	
ALT		
SHIFT		

"Zoom!"

		<b>DIV</b>		<b>MUL</b>	Costume 1	<b>SUB</b>	NAME TARGET
<b>7</b>		<b>8</b>		<b>9</b>		<b>ADD</b>	MASTER TARGET
<b>4</b>		<b>5</b>		<b>6</b>			
<b>1</b>		<b>2</b>		<b>3</b>			
<b>0</b>		.					

	<b>CTRL</b>	<b>DIV</b>		<b>MUL</b>	Costume 2	<b>SUB</b>	SET NAME
<b>7</b>		<b>8</b>		<b>9</b>		<b>ADD</b>	
<b>4</b>		<b>5</b>		<b>6</b>			
<b>1</b>		<b>2</b>		<b>3</b>			
<b>0</b>		.					

	<b>ALT</b>	<b>DIV</b>		<b>MUL</b>	Costume 3	<b>SUB</b>	
<b>Y</b>		<b>8</b>		<b>9</b>		<b>ADD</b>	
<b>4</b>		<b>5</b>		<b>6</b>			
<b>1</b>		<b>2</b>		<b>3</b>			
<b>0</b>		.					

	<b>SHIFT</b>	<b>DIV</b>		<b>MUL</b>	Costume 4	<b>SUB</b>	
<b>7</b>		<b>8</b>		<b>9</b>		<b>ADD</b>	
<b>4</b>		<b>5</b>		<b>6</b>			
<b>1</b>		<b>2</b>		<b>3</b>			
<b>0</b>		.					

<b>F4</b>	<b>DAMAGE</b>	<b>F5</b>	<b>AIM</b>	<b>F6</b>	<b>DEFENSE</b>	<b>F7</b>	<b>NO-HIT</b>
<b>F4</b>		<b>F5</b>		<b>F6</b>		<b>F7</b>	
<b>F4</b>		<b>F5</b>	SAVE BINDS	<b>F6</b>	LOAD BINDS	<b>F7</b>	
<b>F4</b>		<b>F5</b>		<b>F6</b>		<b>F7</b>	

<b>F8</b>	AUCTION	<b>F9</b>	EXIT MISSION	<b>F10</b>	VID SETTINGS	<b>F11</b>	
<b>F8</b>		<b>F9</b>		<b>F10</b>		<b>F11</b>	
<b>F8</b>	MONITOR	<b>F9</b>	QUIT	<b>F10</b>		<b>F11</b>	
<b>F8</b>		<b>F9</b>		<b>F10</b>		<b>F11</b>	