

How to create timer bars with Gnosis

1. Create a new bar (same for cast- and timer bars)
 - Open the Gnosis configuration menu (e.g. type /gnosis into your chatbox)
 - Go to the Castbars sub menu
 - Next to the 'New castbar' button, enter the desired (arbitrary) bar name. Remark: Gnosis will create a globally accessible frame with the entered name. Make sure you press Enter or click the "okay" button after entering the bar name
 - Hit the "New castbar" button
2. Change the newly created bar to a timer bar (it's initially a player castbar)
 - Select the bar by clicking its' name
 - Change 'Bartype' option to 'Multi-Spell Timer'
 - Select the desired default 'Unit' to track casts and/or cooldowns/auras (=dots/hots), etc.
3. Change the look and feel of your bar (this is totally up to you, there are plenty of options to play around with)
4. Configure what your "Multi-Spell Timer" will actually track
 - Go to the 'Hide castbar' sub menu
 - Enter the timer configuration string into the 'Black-/whitelisted spells or timer config' edit box
 - Check out Gnosis' description on curse for more details

Timer configurations

When configuring timer bars, place each command on its own line.

The basic format of each command is **command [options] : type (e.g. spell/item/etc.)**
e.g. **cast unit=target : any**, where *cast* is the command, *unit=target* is one of multiple possible options and *any* is a special keyword to tell Gnosis to display any cast made by the given unit (override *unit=target*).

Timer configuration examples

a) Aura tacking

```
-- lines beginning with two '-' will be ignored (comment)
debuff mine unit=target : Vampiric Touch
not buff unit=player : Power Word: Shield
```

This three liner will display **your** (optional keyword *mine*) Vampiric Touch debuff if it is present on your target. If not, it will remind you to cast Power Word: Shield if it is **not** present on yourself. *buff*, *hot* and *aura help* are all identical commands to display buffs, *debuff*, *dot* and *aura harm* commands to display debuffs. The more general form of *aura* also exists to display both buffs and debuffs of the given name.

Instead of entering the spell name, you can also enter the spell id for the given spell. Gnosis can also track group and raid wide buffs/debuffs by using the commands *groupbuff*, *groupdebuff*, *groupdot*, *groupaura*.

```
-- value of PW:Shield buff on player
aura help auraeffect=100000 unit=player tfs="r<0>" : Power Word: Shield
```

This will display the remaining shield strength of any Power Word: Shield buff, no matter the source (e.g. from other player). *auraeffect=100000* tells Gnosis not to display the buffs duration but its aura effect value. Since Gnosis does not know how strong the effect can be, the user has to pass a sensible maximum value (nothing bad happens if the maximum value is actually too small). *tfs="r<0>"* overrides the time format string, displaying only the remaining shield strength and not its maximum value (with 0 decimals).

For auras with stacks, you can use the optional keyword *aurastacks=max_value*.

b) Conditions

Gnosis allows to activate timer entries based on value and stack count intervals.

The syntax for conditions is *commands [options] : type <conditions>*, where conditions are in the format *<a,b,x,y>*. To be valid, the timer entry has to be in the value range [a,b] (e.g. duration) or in the stack count range [x,y]. Omitting values tells Gnosis to ignore either the upper or lower bound (or both). Values can be entered as absolute values or in percent (e.g. *<10%,50%>*).

```
-- Anticipation charges example (Rogue)
-- stack count >= 4
buff unit=player sbcol="1,0,0" : Anticipation <,,4>
-- no condition needed for < 4
buff unit=player : Anticipation <,,,3>
```

c) Cooldowns

```
-- 13750 is Adrenaline Rush's spell id
cd sbcol="0,1,0,0.8" : 13750 <,10>
cd : Adrenaline Rush
```

This will display the cooldown for Adrenaline Rush. If the remaining cooldown duration is in the interval [0,10] seconds, the status bar color will be changed to green (red, green, blue, alpha). Cooldowns are player based, therefore unit=xyz overrides will be ignored.

d) Resources

```
-- rogue/feral combo points
resource hideicon hidespark
\ mcnt=5 msize=0.9
\ nfs="" tfs="" : combopoints
```

If you ever want to split a command onto more lines, you can do so by using the \ (append token). \ has to be the very first character on the line to append.

- the basic command for this example is *resource : combopoints*.
- *hideicon* and *hidespark* allow to disable icon/spark for each command of a timer bar individually
- *mcnt=x* tells Gnosis to divide up the bar into five pieces, *msize=x.x* tells Gnosis how much space each piece should actually use (where 100% = 1.0 would be maximum size)
- *nfs=""* and *tfs=""* overrides the name and time format strings, in this case disabling any text output

```
-- displays estimated range between player and target
resource unit=target tfs="col<1,1,1>r<0> - t<0>" : range <,4.9>
resource unit=target tfs="col<0,1,1>r<0> - t<0>" : range <,19.9>
resource unit=target tfs="col<0,1,0>r<0> - t<0>" : range <,29.9>
resource unit=target tfs="col<1,1,0>r<0> - t<0>" : range <,39.9>
resource unit=target tfs="col<1,0,0>r<0> - t<0>" : range
```

The configured bar of this example displays the distance between you and your target. I am using this example in a timer with everything hidden but text (fully transparent/no icon/0 border size) and no name format string. Only the time format string is displaying the actual value, which will change its color depending on the estimated distance.

e) Logical operators

Often it is necessary to make sure a condition is true before you even consider showing a cd/buff/etc. This can be achieved by logical combination of multi-spell timer entries.

Gnosis knows three logical operators, **&** (and), **?** (relaxed and) and ***** (or). Logical operators have to be placed as the first character of each command (one command per line), otherwise Gnosis will ignore them.

Always start with either **&** or **?** (mixing the two will yield undefined results) and optionally finish with ***** (or). You can combine as many **&** or **?** as needed.

-- do not try this example at home, since useless!

```
charspec : Combat
& resource : combopoints <4>
& fixed nfs="Eviscerate now!" : Eviscerate
```

Above example shows the usage of the and operator **&**. If our current specialization is Combat (rogue), and we have four or more combopoints, "Eviscerate now!" will be displayed. For this example, Player has to be selected as the default bar unit (therefore no need to override with unit=player).

-- display mana on my priest, combo points on my rogue

```
charspec : Shadow
? charspec : Combat
? resource portrait=player nfs="who - name" : combopoints
* resource portrait=player nfs="who - name" : power
```

In this example we are using the relaxed and operator **?**. Only one charspec has to be true (displayable bar if used as stand alone) for the resources to be displayed. On my priest, *resource : combopoints* is not displayable, therefore *resource : power* will be evaluated due to the or operator *****. On my rogue, resource combopoints will be displayed. For this example, Player has to be selected as the default bar unit (therefore no need to override with unit=player).

- f) Counters in combination with logical operators (the given example looks kind of complex but is quite simple, honestly ☺)

Following example will tell two groups to make ready for interrupting Fungal Flesh Eaters casting Decay:

```
-- Brackenspore: no Fungal Flesh-Eater?
not npc unit=boss2 stopcnt="brack1" : 79092
& exit stopcnt="brack2" : 0
-- show Decay cast
cast unit=boss2 startcnt="10-brack_change" : 160013
-- swap kick #1/#2
counter : brack_change
& counter stopcnt="brack_change" : brack1
& exit stopcnt="brack1" startcnt="240-brack2" : 0
counter : brack_change
& counter stopcnt="brack_change" : brack2
& exit stopcnt="brack2" startcnt="240-brack1" : 0
-- display Kick #1/#2 status
counter : brack1
& fixed nfs="Prepare: Group #1" : 1766
counter : brack2
& fixed nfs="Prepare Group #2" : 1766
-- start counter when boss2 is a Fungal Flesh Eater
npc unit=boss2 startcnt="240-brack1" : 79092
& exit : 0
```

- if boss2 is no Fungal Flesh-Eater (npc id 79092, e.g. create a Gnosis timer to display the current target's npc id with "npc unit=target : any"), all counters will be stopped
- if boss2 is casting Decay, the bar will display the cast; also, counter 'brack_change' will be started
- if counter brack_change is running, brack1 or brack2 will be started, depending if brack2 or brack1 are already running
- the bar will display which group will have to kick next, depending on whether counter brack1 or brack2 is running
- if every other entry failed, counter brack1 will be started if boss2 is a Fungal Flesh Eater
- the used counter durations of 240 and 10 seconds are totally arbitrary (and could be a lot shorter)
- counters always start at 0s and end when they reach the given duration (even though, when displayed, remaining duration is shown). This is important to know if you use conditions in combination with counters (see b) Conditions)

Remark: restarting a counter is possible by using stopcnt and startcnt in one command.

g) Global cooldown indicator

You can create a global cooldown indicator by creating a new castbar and selecting **GCD Indicator** or **GCD Indicator (non casttime spells)** as the targeted unit.

In case you want to displays either castbar or global cooldown indicator, you can

- use the **non casttime spells** version of the indicator, which has its drawbacks since Gnosis does not know about buffs, etc. which might make a spell instant
- create a timer bar with **gcd hidden="exact player castbar name" : any** as timer configuration, which will show all global cooldowns as long as the player castbar is not shown

For an example, you can always grab the import strings for my shadow priest example configuration (see link below) which uses the timer bar gcd feature.

For more information and more examples, please visit Gnosis website @ curse gaming: <http://www.curse.com/addons/wow/gnosis>