

Macro	Description
#attrmod[attribute_id]	Returns the final modifier for the specified attribute.
#appenddesc[thing_id, desc]	Add to the description of the designated thing.
#applybonus[type, pick, bonus]	Allows you to add a non-stacking bonus to any pick with General Modifiers. Intended for use where modifier bonuses do not stack with each other - this is the case for Alchemical, Competence, Insight, Luck, Morale, Sacred and Profane bonuses. This means if you add two Insight bonuses only the best will apply.
#applydr[type, bonus]	Allows you to add a type of damage reduction to the hero. Using this macro makes sure that the highest DR value is used, since DR doesn't stack.
#applypenalty[type, pick, bonus]	Allows you to add a non-stacking penalty to any pick with General Modifiers. Intended for use where modifier bonuses do not stack with each other - this is the case for Alchemical, Competence, Insight, Luck, Morale, Sacred and Profane bonuses. This means if you add two Insight penalties only the worst will apply.
#applyresist[type, bonus]	Allows you to add a type of damage resistance to the hero. Using this macro makes sure that the highest resistance value is used, since resistance doesn't stack.
#applysr[bonus]	Allows you to add spell resistance to the hero. Using this macro makes sure that the highest SR value is used, since SR doesn't stack.
#applyvalue[type, bonus]	Allows you to add a value to an ability. Using this macro makes sure that the highest value is used. If this value can be stacked with other things that increase it, you should use the #value[] macro instead.
#applyvalue2[type, bonus]	as #applyvalue[] for abValue2
#applyvalue3[type, bonus]	as #applyvalue[] for abValue3
#applyvalue4[type, bonus]	as #applyvalue[] for abValue4
#attrbonus[XXXX]	use this to retrieve the final bonus for an attribute (if a character has a negative attribute modifier, they have an attribute bonus of 0).
#attrbonus1[XXXX]	use this when a number of uses is set equal to an attribute bonus, and the number added can't be less than 1.
#attrvalue[attribute_id]	Returns the final value of the attribute specified.
#competencebonus[pick, bonus]	Allows you to add a competence bonus to any pick with General Modifiers. Competence bonuses do not stack with each other, so if you add two competence bonuses only the highest will apply.
#CPA[id]	Deprecated - use #PP[]
#dc[ability_id]	The abDC field stores the DC of the ability. If abDC is greater than 0, it is appended to the livename - the version of the ability's name that is displayed on the special tab and printouts.
#duration[ability_id]	The abDuration field stores a value that represents the duration an ability will last for. Generally, this is only used when the duration of the ability is specific length of time, and the ability is either on or off. If the ability can be used a few rounds at a time, the duration should be stored in trkMax.
#enhancementbonus[pick, bonus]	Allows you to add an enhancement bonus to any pick with General Modifiers. Enhancement bonuses do not stack with each other, so if you add two enhancement bonuses only the highest will apply.
#extracrit[target weapon,damage text,source's name]	Using this macro is the same as using the #extradamage macro, but it adds its effects to the critical hits instead of the damage.
#extradamage[target weapon,damage text,source's name]	Additional damage can now be applied to a chosen weapon. target weapon is where you tell Hero Lab what weapon you'd choose. The code you'll use here is the same as in if you wanted to use the #applybonus[] macro to apply a bonus to the same pick.damage text is the text you want to add, for example; "+1d6 fire" or " plus grab".source's name is the name of the ability that's applying the effect. You'll almost always use field[name].text for this.
#Fame[id]	Returns the fame of a faction
#featlevelcount[]	Works like levelcount[] but it counts all the classes that "count as" the specified class as well.
#hasability[ability_id]	Checks whether the specified ability is present and active (reached the correct level and not disabled by something else) for the purpose of pre-requisites. This returns 0 if the ability is not active, and 1 if the ability is active.
#hasarchetype[]	Checks whether the specified archetype is present for the purpose of pre-requisites. This returns 0 if the archetype is not present, and 1 if the archetype is present.
#hascondition[]	Checks whether the specified condition is present for the purpose of pre-requisites. This returns 0 if the condition is not present, and 1 if the condition is present.
#hasdeity[]	Checks whether the specified deity is present for the purpose of pre-requisites. This returns 0 if the deity is not present.
#hasfeat[feat_id]	Checks whether the specified feat is present for the purpose of pre-requisites. This returns 0 if the feat is not present, and 1 if the feat is present, or if a feat which "counts as" that feat is present.
#hasrace[race_id]	Checks whether the specified race is present for the purpose of pre-requisites. This returns 0 if the race is not present, and 1 if the race is not present.
#hassubtype[subtype_id]	Checks whether the specified subtype is present for the purpose of pre-requisites. This returns 0 if the feat is not present, and 1 if the subtype is present.
#hastrait[trait_id]	Checks whether the specified trait is present for the purpose of pre-requisites. This returns 0 if the feat is not present, and 1 if the trait is present.
#hastype[]	Checks whether the specified typeis present for the purpose of pre-requisites. This returns 0 if the type is not present, and 1 if the type is present.
#hasvanity[id]	checks vanity
#levelcount[class_id]	Counts how many levels you have of the specified class.
#makeclassskill[skill_id]	Allows you to designate a skill as a class skill for this character.
#makespecskill[id]	simplify the scripting for making a skill a specialized skill.
#PP[id]	Returns the Prestige Points of a faction.
#situational[chosen save,bonus text,source's name]	chosen save is the chosen save. bonus text is the text you want to add to that saving throw, and source's name is the name of the ability that's applying the effect. You'll almost always use field[name].text for this.
#skillbonus[skill_id]	Allows you to add a generic bonus to any skill
#skillinnate[skill_id]	sets the innate skill on a race
#skillranks[skill_id]	Allows you to retrieve the number of user ranks in any skill.
#specvalue["XXX"]	returns the value of a class special. Note that the quotes around the ID of the class special are required.
#totallevelcount[]	Counts how many total class levels you have.
#TPA[ID]	Deprecated - use #Fame[]
#trkmax[tracker_id]	The trkMax field is the number of charges/uses/rounds/whatever available to a thing.
#value[ability_id]	The abValue field stores a value. What this value means differs from ability to ability.
#value2[ability_id]	The abValue2 field stores a second value, in addition to the one stored in abValue. What this value means differs from ability to ability.
#value3[ability_id]	The abValue3 field stores a third value, in addition to the one stored in abValue. What this value means differs from ability to ability.
#value4[ability_id]	The abValue4 field stores a fourth value, in addition to the one stored in abValue. What this value means differs from ability to ability.
#resmax	change the max of a resource
#racialbonus[]	add racial skill bonus
#racialsituational[]	add racial situational bonus
#BAB[]	returns Base Attack Bonus
#cancastdivine[]	used for prereqs of minimum divine spell level
#cancastarcane[]	used for prereqs of minimum arcane spell level

	--Helper Tags--
Helper.AlwaysMast	Forces armor to always be masterwork
Helper.BonFtNoPre	For class
Helper.DamageUp	Increase Natural Weapon Damage Die size by one - Checked Post-Levels 10000
Helper.DamageDown	Decrease Natural Weapon Damage Die size by one - Checked Post-Levels 10000
Helper.NatPrimary	Make the natural attack a "Primary" attack.
Helper.NatOverSec	Makes the natural attack into a secondary attack.
Helper.NatTwoHand	Forces the weapon to be treated as a two-handed weapon or 1.5str modifier.
Helper.NatSizeDmg	Tells HL to change the die size based on the current size of the character. When using this set the default damage as if the creature was size Medium.
Helper.NatIgnWeap	bootstrapping a Natural attack with this tag renders it immune to the effects of having a weapon equipped. That is, it will remain a primary attack with weapons.
Helper.TwoWpLight	weapons with this tag count as Light for two weapon fighting but not for any other purpose.
Helper.NoAttacks	for specials which do not make attacks but nonetheless should be shown on the melee line of the statblock/on the weapons tab. This is used for the Swarm attack special ability.
Helper.NoHalfStr	weapons with this tag do Str modifier damage even when equipped in the off hand. This is similar to Double Slice but can be applied to only certain weapons (rather than globally to all weapons). It is currently used for Monks with their Unarmed Strikes.
Helper.NoHandWeap	when added to a weapon, it will no longer count as taking a hand to wield (thus preventing conflict with other equipped weapons). This is used for Barbazu Beards, Bladed Boots, and several new weapons from the Advanced Race Guide.
Helper.NoCureInf	when assigned to a hero it prevents the bootstrapping of the bonus Cure/Inflict spells known for an Oracle. Used for the Purifier oracle archetype.
Helper.BrdPerform	which marks the Bardic performance abilities so that they can be more easily selected in foreach/custom expressions.
Helper.ShowFlurry	This makes the Monks Flurry of Blows display for the weapon.
Helper.Improvised	Marks improvised weapons so they can be found and manipulated.
Helper.HalfProfMe and Helper.HalfProfRn	tags have been added, it causes a weapon to not take nonproficiency penalties for melee or ranged attacks, respectively.
Helper.NoArmor	When added to Armor will set the armor bonus to 0. Partial way to disable armor on a character.
Helper.MatForceMs	Forces Materials to make the Gizmo Masterwork
Helper.NoSelect	???
Helper.Displace	???
Helper.NoPick	???
Helper.TrainOnly	Marks skills that can only be used trained, delete to use untrained (example Bard Jack of all trades: perform eachpick.delete[Helper.TrainOnly])
Helper.ClassSkill	Marks skill as class skill
Helper.Proficient	Added to weapon you are hero is proficient with
Helper.MatRqClLea	Must be made of cloth or leather
Helper.Shadow	Used to make items show that are bootstrapped to item powers
Helper.SourceEqp	Used when items need to detect if their source is equipped beyond normal conditional
Helper.ClsCastLev	SLA Caster Level based on Class level
Helper.HDCastLev	SLA Caster Level based on Class Level and racial HD
Helper.TrCastLev	SLA Caster Level based on Mythic Tier
	--Hero Tags--
Hero.MutExArmor	when added to the hero it will increase the natural armor bonus granted by a mutagen by 1 (per tag assigned).
Hero.ProfImpMel and Hero.ProfImpRng	tags have been added, granting proficiency in improvised melee and ranged weapons respectively.
Hero.TwoWepHalf	When applied to the hero, it halves any penalties to attack from two weapon fighting (used for the Thunderstriker fighter archetype).
	--Hide Tags--
Hide.Weapon	This will hide a weapon and not have it be displayed.
	--Misc Tags--
thing.skippreq	Skips prereqs on feats
	--Tag Groups--
ClsAllowSp.?	assign tag to class helper to add spell to spell list (? = spell ID)

ClsDenySp.?	Like the CIsAllowSp tags, these are applied to a class to modify their allowable spells, but instead of adding a spell this tag group removes it instead.
SpellEff.?	Allows feats and effects that depend on a certain effect (like the Elemental Focus feat needing appropriate elemental damage) can identify which spells they apply to.
DamageOver.?	When added to a weapon, that weapon will use another attribute in place of Strength to determine its damage bonus.
DamageOpt.?	if you want to use the higher of Str or the new attribute for damage.
	--Weapon Tags--
wMain.1_1	1
wMain.1d2_2	1d2
wMain.1d3_3	1d3
wMain.1d4_4	1d4
wMain.1d6_5	1d6
wMain.1d8_6	1d8
wMain.2d4_6	2d4
wMain.1d10_304	1d10
wMain.1d12_404	1d12
wMain.2d6_104	2d6
wMain.3d4_104	3d4
wMain.2d8_204	2d8 -> 2d10
wMain.2d8_305	2d8 -> 2d8
wMain.4d4_204	4d4
wMain.3d6_8	3d6
wMain.2d10_205	2d10 -> 4d8
wMain.2d10_504	2d10 -> 3d10
wMain.2d12_604	2d12
wMain.3d8_306	3d8 - > 4d8
wMain.4d6_106	4d6
wMain.3d10_505	3d10
wMain.5d6_106	5d6
wMain.4d8_206	4d8 -> 6d8
wMain.6d6_107	6d6
wMain.4d10_506	4d10
wMain.7d6_107	7d6
wMain.4d12_606	4d12
wMain.6d8_207	6d8
wMain.8d6_108	8d6
wMain.5d10_507	5d10
wMain.9d6_108	9d6
wMain.7d8_207	7d8 > 8d8
wMain.10d6_11	10d6
wMain.6d10_507	6d10
wMain.8d8_208	8d8
wMain.8d10_508	8d10
wMain.10d10_509	10d10
wMain.12d10_510	12d10
wOff.?	Same as wMain tags
wProfReq.Simple	Simple Weapon Prof required.
wProfReq.Martial	Martial Weapon Prof required.
wProfReq.Exotic	Exotic WEapon Prof required.
wProfReq.Firearm1	One-Handed Firearm Prof required.
wProfReq.Firearm2	Two-Handed Firearm Prof required.
wProfReq.Siege	Siege Prof required.
wProfReq.SiegeFire	Siege Firearm Prof required.
wProfReq.SiegeLt	Siege (Light) Prof required.
wProfReq.ImprovMel	Improvised Melee Prof required.

wProfReq.ImprovRng	Improvised Ranged Prof required.
SkillOver.?	Change the ability score a skill uses
SkillOpt.?	Use better of default ability score or selected score
wAttMelee	field, which adds to melee attacks. This stacks with the wAttBonus field (which adds to both melee and ranged attacks).
Helper.NoAttrDam	prevents any damage bonus from the active attribute
	-Armor Check Penalty Tags=-
ImpArChInf.?	tags are skills, applied to Hero to remove armor check penalty from specific skill
ImpArmChk1.?	tags are skills, applied to hero to reduce ACP by 1
ImpArmChk2.?	tags are skills, applied to hero to reduce ACP by 2
ImpArmChk3.?	tags are skills, applied to hero to reduce ACP by 3
ImpArmChk4.?	tags are skills, applied to hero to reduce ACP by 4
ImpArmChk5.?	tags are skills, applied to hero to reduce ACP by 5

Ability or HasAbility	
Damage Reduction, Aligned	abDamRdAli
1 Hit Point	fcHP
1 Skill Point	fcSkillPts
Burrowing	xBurrow
Climbing	xClimb
Damage Reduction	xDamRd
Damage Reduction, Adamantine or Bludgeoning	xDamRdAdB
Damage Reduction, Adamantine	xDamRdAdam
Damage Reduction, Melee or Magic	xDamRdArro
Damage Reduction, Bludgeoning	xDamRdB
Damage Reduction, Bludgeoning or Magic	xDamRdBMg
Damage Reduction, Bludgeoning or Piercing	xDamRdBP
Damage Reduction, Cold Iron or Evil	xDamRdCIEv
Damage Reduction, Cold Iron or Silver	xDamRdCISi
Damage Reduction, Chaotic	xDamRdChao
Damage Reduction, Epic	xDamRdEpic
Damage Reduction, Evil or Magic	xDamRdEvMa
Damage Reduction, Evil or Silver	xDamRdEvSi
Damage Reduction, Evil	xDamRdEvil
Damage Reduction, Cold Iron or Good	xDamRdGoCI
Damage Reduction, Good or Lawful	xDamRdGoLa
Damage Reduction, Good or Piercing	xDamRdGoPi
Damage Reduction, Cold Iron	xDamRdIron
Damage Reduction, Cold Iron or Lawful	xDamRdLaCI
Damage Reduction, Lawful	xDamRdLaw
Damage Reduction, Lethal	xDamRdLeth
Damage Reduction, Magic	xDamRdMag
Damage Reduction, Magic or Silver	xDamRdMgSi
Damage Reduction, Magic or Slashing	xDamRdMgSl
Damage Reduction, Slashing	xDamRdS
Damage Reduction, Slashing or Bludgeoning	xDamRdSB
Damage Reduction, Silver	xDamRdSil
Damage Reduction, Vorpal	xDamRdVorp
Damage Reduction, Wood	xDamRdWood
Damage Resistance, Acid	xDamRsAcid
Damage Resistance, Cold	xDamRsCold
Damage Resistance, Electricity	xDamRsElec
Damage Resistance, Fire	xDamRsFire
Damage Resistance, Negative Energy	xDamRsNeg
Damage Resistance, Positive Energy	xDamRsPos
Damage Resistance, Sonic	xDamRsSoni
Flight	xFly

Fortification	xFortifica
Spell Restiance	xSplRs
Swimming	xSwim

~ If we're disabled, do nothing
doneif (tagis[Helper.SpcDisable] <> 0)
~ If we're not shown, just get out now
doneif (tagis[Helper.ShowSpec] = 0)
~ only run the rest for the first copy
doneif (tagis[Helper.FirstCopy] = 0)
~ If we're disabled, do nothing
doneif (tagis[Helper.FtDisable] <> 0)
~ If we're not enabled, get out now
doneif (field[plsOn].value = 0)
~ If nothing chosen, get out now
doneif (field[pChosen].ischosen = 0)
~ If we're not enabled, get out now
doneif (field[abilActive].value = 0)
~ If we're not equipped, get out now
doneif (field[glsEquip].value = 0)
~ If we're not chosen, get out now
doneif (field[usrChosen1].ischosen = 0)
~ If we have no more grit then get out now
doneif (hero.childfound[trkGrit].field[trkLeft].value = 0)