Dangerous Journeys: Mythus Character Sheet

Persona Sheet: This sheet is designed for use with major characters, notably Heroic Personas or any other Personas (i.e. major antagonists) for which detailed information is required—all K/S areas and sub-areas listed, all Castings listed, equipment, armor detailed, etc. Most fields are self-explanatory if you are familiar with the DJ:Mythus rules, but I’ll try to discuss quirky or oddball fields here.

Name and Vocation are informational only; Vocation TRAIT has the potential to be used by calculations, etc. but at this point that is not the case. SEC is informational; possibly another field will be added to allow an optional “alternate” SEC (for Primitive Vocation personas, for instance), but not at this time. Attractiveness *should* generally be numeric, but the field does allow text to add context to the Attractiveness value (such as information about creatures and if their “extreme Attractiveness” will provoke necessary die rolls to avoid the effects thereof). Another field exists to list inner Beauty or Inner Ugliness.

If the Persona is in the Dazed state, DAZED! will be present in bold red right under the Mythus header. This will flag automatically when Mental or Spiritual damage exceeds the EL of the appropriate TRAIT or Physical damage exceeds the WL, and will go away when the damage levels have gone back down (below EL for Mental/Spiritual; below RL for Physical). Because there are other ways of gaining the Dazed condition, a checkbox labeled Dazed (manual)? is available to manually set the Dazed condition for those other situations. Note that being Dazed should increase Difficulty Rating of most things, but at this point that is not programmed into the sheet.

Move is computed from P TRAIT, augmented by armor SF Penalty and the effects of being Dazed. The various movement rates are listed in the standard “yds/BT”, and the various standard adjusted movement rates (like for Running or Cautious movement) are also given for reference.

AP/Gs is where to place your unused (General) Accomplishment Points, Joss is for any Joss Factors the Persona possesses.

Heka is a listing of current Heka (modifiable) along with a computation of the Persona’s Max Heka (non-modifiable, at least directly). Max Heka is computed using the standard Core rules, modified slightly by the “official” rule clarifications from Mythus Masters Magazine #3. Since there is some discrepancy in the rules themselves as to how to compute Heka, there are several options available on the Config Tab (see below) which can be selected to modify those behaviors. If you’re doing something here which is not covered by the options on the Config page, let me know and I may consider adding that option to the computation algorithm, but I’m more likely to just add a field to the Config Tab which will let you stuff it with more Heka should you so choose.

Heka Aperture is computed using the rules from Journeys magazine, reprinted in Mythus Masters Magazine #2, for Heka “Apature”. I have gone out on a limb and assume it was a typo instead of a new form of magickal terminology, but if you have heartburn over this you are more than welcome to create your own Character Sheet. 😊

Any limitations on how Heka can be used (such as Astronomy Heka only being used for Astrology castings) should be noted in the Heka Limitations text box.

Next, the details start in earnest. There are four tabs which contain the bulk of Persona data, and one tab, Config, which contains configuration information as well as rules options used.

The Main Tab:

The Main Tab contains the critical set of stats for the Persona.

Characteristics:

These include all the ATTRIBUTES (Cap, Pow, Spd), all the CATEGORIES computed from those, and the TRAITS computed from those. The latter two are not modifiable by the user… they’re taken from the ATTRIBUTES, which *are* modifiable. The d20 next to each ATTRIBUTE, CATEGORY, or TRAIT rolls a standard K/S Skill check against each of those when clicked, querying the user on which Difficulty Rating to apply.

Avoidance for Mental, Physical, and Spiritual activities are listed as well, with rolls available, and Dodging Factor, if any, is listed. These are, once again, computed, non-modifiable values at this point.

Weapons:

Note the Import New checkboxes available here. They open a hidden text field and button where you can paste in weapon stats in a standardized format that you can find in an associated Excel spreadsheet (see Links, below). After pasting and clicking the Import button, the weapon should automagically appear if all has gone well. Note that this is not in any way required: you can type in all these stats manually.

These are divided into Melee and Missile categories, with a summary of the most critical stats visible, as well as the BAC and Damage rolls directly available. To add a new weapon, click the +Add button located below the Weapon area, and fields will appear for you to fill out. The Modify button will allow you to reorder or delete weapons.

Clicking on the checkbox to the left of any weapon opens a more detailed list of stats for the weapons. (These need to be labeled better by me. Sorry. I’ll do that eventually. For now, hovering over a field will give you its name.)

The weapon name is self-explanatory.

The next field is the Sub-Area of a combat K/S which that weapon belongs to. All Sub-Areas from the Core Rules are available in the pulldown, again supplemented by the “official” corrections as found in Mythus Masters Magazine #2. This is used to compute BAC, so it’s kind of critical to get it right.

WP, or Weapon Points, are used to compute BAC as well.

Next is composition, in this case Metal or Wood or Combination. This is not currently used, but could be when developing Parrying macros in the future, since that is where it is relevant.

SF, or “Speed Factor”, is next, and this can be used for Initiative computations.

NumAtks is the base number of attacks for that weapon. All melee and hand-to-hand weapons have a 1 for this; missile weapons will often have a Rate of Fire or ROF parameter, and that’s what should be entered here. Note that the sheet will auto-compute the actual number of attacks per CT from this NumAtks/ROF parameter as well as the appropriate Combat STEEP value, just enter 1 for melee/HTH and ROF for missiles and the sheet will do the rest.

Damage type is next, with Piercing, Blunt, and Cutting being most common. Mixed damage types are available where appropriate.

Next is Damage, where the base amount of damage should be listed. Note that Combat HTH weapons like Hand, Feet, Nunchakus, etc. will have this auto-populated by the sheet according to STEEP value, but all other weapons will need this filled in. The sheet will auto-calculate any modifiers to damage from high STEEP or PMPow/PNPow as well.

Weapon Reach is informational, and is in yards. WeaponBAC is a computed value—if you change it, it will change back. I may make it invisible in the future.

Weapon DurHard and WeaponDurHP are the durability numbers. They are not currently used, but when Parrying is fully implemented they will be important factors along with Weapon Composition.

Weapon Special is a text field for all those footnotes in the Core rules for each weapon, or for describing any Heka-enhancements to the weapon beyond just better BAC and damage and speed factor.

WeaponMiscDmg is for any adders to damage from Heka-enhancement or for any other reason; WeaponDmgTot is, like WeaponBAC, a computed quality which will likely be made invisible at some point. If you change it, the changes won’t stick.

WeaponMiscBAC is for any adders to BAC from Heka-enhancement or some other reason.

WeaponNumAtksComp is the computation of how many attacks one gets with this weapon. Changes to it won’t stick, and it may go invisible in the future.

For missile weapons, the Point Blank, Short, Medium, Long, and eXtreme ranges are listed.

Armor:

As will weapons, Armor can be pasted in with the Import New functionality, or directly entered into fields.

Clicking Add+ will allow you to enter a new Armor item, the Modify button will let you reorder or delete them.

You can type in the protection that each armor item offers to each type of damage as well as the Speed Factor Penalty if any; further, you can check the boxes for which areas the armor covers (Nonvital, Vital, Supervital, and Ultravital are N, V, S, and U).

The number of hits for armor should be 10 in general, and can be tracked and reduced as armor is damaged per the rules.

All armors will be totaled in the display below, so totals for NonVital, Vital, SuperVital, and UltraVital will be available for every damage type listed. In addition, Speed Factor Penalties from all Armor Items will be totaled and displayed, for use in Initiative, Dodging, and Movement.

The Knowledge and Skills Tab

This is where all the important information about the K/S Areas and Sub-Areas are stored. To keep things clean, it only displays K/S Areas which have been selected in the Config Tab (see below).

It lists which TRAIT is associated with a given K/S Area (unmodifiable), the STEEP (modifiable here), how many sub-Areas the STEEP allows, and any Heka generated from the K/S area. Next, there is a button which runs a K/S Skill Check against that K/S Area. Optionally, if the K/S Area is associated with Castings, the Max Casting Grade will be listed.

Clicking on the checkbox near the number of Sub-Areas opens a subdirectory which lists all Sub-Areas, with checkboxes for Has (H), Specialized (S), and Proofed (P). Has indicates the Sub-Area is possessed, Specialized indicates that the Persona specializes in this Sub-Area (giving 1.5x the STEEP in most cases and 0.5 STEEP to all non-specialized Sub-Areas) and Proofed indicating that these non-specialized Sub-Areas are now to be given 1x STEEP, once the K/S STEEP has increased to the point where all Sub-Areas are gained. Little to no error checking is done here, so may the JM beware! The associated button makes a K/S Check against that Sub-Area.

Clicking on the checkbox next to Heka generated will show a detail of how that Heka is computed, including STEEP, any multiplier to STEEP for Full Practice, Vow, etc., and extra Heka generated by associated ATTRIBUTE, CATEGORY, or TRAIT. The Misc field is mostly unused “just in case”, and may go away in the future. It is only used in a very few cases, and will have no effect in other cases. If it remains, it will be as a way for people to add in extra Heka generation which is not otherwise covered by the rules.

Several Areas and Sub-Areas have associated text fields (languages, mostly) which can be filled out; in cases where more than one of these is permitted (Foreign Language, Economic Mercantilism Sub-Area, StreetWise subcultures) Add and Modify buttons are present which allow multiples to exist. In a similar vein, Weapons Special Skill allows for selection of multiple combat K/S sub-Areas for each “trick”. Note that again there is no error-checking: though the sheet will tell you how many you SHOULD have, per the rules, you are free to fail to live up to this and the sheet won’t stop you.

Note that the Areas and Sub-Areas don’t always match the Core rules, as they have been augmented with the “official” changes in Mythus Masters Magazine #1.

Note also that Psychogenics (Mental, Physical, and Spiritual) are available as K/S Areas even though their inclusion in Mythus is debatable: they are referred to in the Core rules as possessed by Non-Human Personas as a K/S Area, but in other works, like Mythus Magick, it is clear that Psychogenics are supposed to be present in other milieux only. Use at your own risk, and see the Config Tab optional rules for details.

The Misc Tab

A variety of different text fields to record all the squishy, colorful information that makes a character more than just a collection of stats. None of these has a Roll20 effect. Enjoy them!

The Powers/Castings Tab

Powers are for “Heka-Engendered Powers” and, really, any other special abilities from Quirks or otherwise, that need to be detailed in game usable fomat. The Import New functionality exists (see Weapons, Armor, Castings), but as this is somewhat infrequently used for Personas mostly it is likely the Add+ button will be used instead. The power’s name and its use frequency will be displayed by default, but clicking on the checkbox next to the Power name will allow editing of the fields like duration, area, and detailed descriptions.

Castings are slightly more complicated.

First, again, Import New is available for cutting/pasting Castings into the sheet (someone should compile a spreadsheet of Castings in this Character Sheet’s format—hint, hint…). It can accept multiple Castings at once, but I have not tested the limitations of this (I don’t know if a glommed together list of 100 Castings will break the importer, for instance. Let me know what you find.)

There will be a tab for whatever K/S Areas are possessed by the Persona which have associated Castings, and clicking on that tab will bring you to a list of Castings, by Grade, as well as some information on STEEP for that K/S and Activation Cost Energy aka Base Heka Cost for each grade. Also listed is the associated Difficulty Rating for each Grade (this is automatically computed using STEEP and Full Practice status (see Config Tab), below).

Each Grade has the familiar Add+ and Modify buttons with which to manually type in or organize/delete Castings. At the top level, the Castings list the name of the Casting and the Casting time it is associated with, i.e. Eyebite, Charm, Cantrip, etc. There are three checkboxes meant to denote Known, Recallable, and Studyable Castings, but they are currently poorly marked. The d20 icon next to each casting will send all the Casting’s information to the Chat window, and if you include inline rolls in any of the fields it will do things like roll damage for you automatically. (For instance, if you write “This is a crazy powerful spell. Terrifying, doing [[3d20]] damage” in the E/F/M field for the Murderhobo Cantrip, any time you click on the Die icon next to that Casting it will roll 3d20 and put the resulting value in the middle of that sentence in the Chat window.) It will also apply the Grade-appropriate Difficulty Rating to your STEEP and make a K/S Check for the success or failure of the Casting, and report that in the Chat window. It will NOT reduce your current Heka, and probably never will. You’ll have to do that manually or write an API for it. Sorry. There are just too many implications for me to do that in any robust way.

Clicking on the checkbox to the left of the Casting name will open the details of the Casting… Time, Area, Heka costs, etc.

Note that how Castings are organized may be getting redone soon. I will make every effort to make it backwards compatible, but I’m not currently sure if that will work out or not.

The Config Tab

This tab contains lots of critical “set once and forget it” bits, as well as some that might be accessed in the course of play, but only rarely.

It is possible to use the Import New Persona text box to import an entire persona, with stats, K/S areas, and STEEP using the format from the Excel file in the Links section below. It will not currently import weapons, armor, Powers, or Castings all at once with the rest of the character, though that’s on the roadmap; you’ll have to do multiple steps to get everything. This is more an experimental tool so far, meant to entice someone into coding a character generator that will hopefully spit data out into this format. Note that Roll20, for legal reasons, forbids me from doing this internal to the sheet, but we are allowed import functionality. It will serve, I hope.

Display Heka Components will give you an idea of what’s under the hood when it comes to computing Heka Max. It’s informational only, though I might put in a generic field so rulebreakers and other blessed geniuses can have some means of boosting Heka above what the rules generally permit.

ShadowAttrs will go away soon; it’s a debug tool for me mostly.

Ah… Variant Rules… I’m not going to go into too much detail on the canonicity of the rules here; where there is ambiguity in the core rules which was not cleared up in Mythus Masters Magazine, I have generally allowed for both where it’s easy to code and picked a side where it was not. Current rules variations permitted are:

HekaRules Version:

In the Core Rules and Mythus Magick there is a table which lists each Heka-generating Area and its associated ATTRIBUTE/CATEGORY/TRAIT of additional Heka which should be added in. That table sometimes lists two Attributes added together for the additional Heka.

However, if you read the individual K/S Area descriptions in the Core Rules, any time two Attributes are listed for Heka generation, they are not added together but averaged. This is kind of a conundrum. (There are also several cases where the K/S descriptions identify completely different Attributes than the table, and I have gone with the table in most (all?) cases.

At any rate, HekaRules version lets you decide on adding or averaging ATTRIBUTES when multiples are used. The former gives less Heka, the latter gives slightly more. JM option.

Perception Type for Attacks: There is some textual evidence that the BAC computation could use either Mental or Physical Perception in its bonus calculation, though I come down on Physical. Nevertheless, I leave it as an option for Personas with both to choose one or the other. JM option.

Dodging Armor Rule Options: It’s never really spelled out in the core Rules how to use Armor SF Penalty; I have assumed you reduce your Initiative with it, and use it in my Initiative checks. What is clearer is that you are supposed to modify your Dodging Factor with it, but the Core rules say all SP Penalties should be applied to Dodging Factor, while Mythus Masters Magazine provides an “official” correction that it should only be SF Penalties above 5. I have included both as options. JM decision.

Psychogenics Heka Computation Rules: Challenging and controversial. Longer than I want to get into here, but I have provided two ways of computing Psychogenics Heka. One is strict according to the Dangerous Journeys: Unhallowed rules—the only ruleset where the details of Psychogenics were laid out in any detail—and uses an alternative form of energy known as Vril and then multiplies that x5 to get Heka. I compute Vril here the way the Unhallowed rules tell you to (using all 6 \*Pow statistics), and then multiply it by 5 to generate Heka. Technically there’s supposed to be an exception here for Physical Combat Psychogenics Sub-Areas, where their Vril is only multiplied x2 to obtain Heka, but I cut corners. Sorry. The alternative I have used here is to treat each Psychogenics as a normal Heka-generating K/S Area (like Astrology or Heka-Forging) with Heka adders of the Physical \*Pows for Psych.(P), the Mental \*Pows for Psych.(M), and the Spiritual \*Pows for Psych.(S). I think that’s in keeping with the spirit of Psychogenics being powered by the \*Pows Vril computation while keeping them firmly grounded in the Mythus Heka-computation “STEEP + ATTRIBUTE” method. I am open to suggestions on other methods, and if they’re not insane or difficult to code I will happily incorporate them as options. This is what we have with a contradictory ruleset. JM option.

TRAIT Heka is checkboxes which define whether the Persona possesses Heka from their Vocational TRAIT or more TRAITs, as the rules allow.

# Heka Generating K/S Areas are informational, though they are used for Heka Aperture computation and for whether Heka-Forging gets full Physical TRAIT Heka as an adder.

Heka Multipliers are for Full Practice Priestcraeft and Dweomercraeft. Vocational school should be selected, if present; if FP is checked, Vocational School or Ethos or both MUST be selected or Bad Things might happen to Heka computation.

Finally, there’s K/S Areas Possessed. This determines which K/S Areas are present on the Knowledge and Skills Tab. A check in the first checkbox, “C”, makes the K/S Area appear in the K/S Tab. A lack of checkbox means it won’t show up. The “H” checkbox is whether that K/S Area will generate Heka or not. By default, when a character is started, all the Heka-generating areas per the rules will have this checkbox selected. Areas which are contingent on another K/S Area like Astronomy or Musical Composition will automatically “self-check” when necessary, and other areas which generate Heka due to Vocation (like Ecology/Nature Science for Wiseman/Wisewoman) or due to house rules can be checked as necessary.

The last checkbox on the Config Tab is that to switch to the Bestiary Sheet. Most of the stats will transfer over directly, but there are exceptions and they will be covered in the next section.

Bestiary Sheet: This sheet is an abbreviated form of stats, mostly geared toward the level of details necessary in combat and perhaps to various activities the target persona or creature might perform with or against the Heroic Personas. It follows the format of the Mythus Bestiary book, with little touches of the stats blocks in the Core rules and the Necropolis adventure setting. Most of the stats are the same, but I’ll touch on the differences here.

First to mention: to switch from the Bestiary Sheet back to the Persona Sheet, scroll all the way to the bottom of the Bestiary sheet to the checkbox marked Switch to Persona Sheet. All transferable stats will be there, and vice versa when going the opposite direction.

Also, as with the Persona Sheet, an entire Bestiary Sheet entry can be pasted into the Import New Critter text box if the syntax is right (see Excel template in Links).

Next, not all settings are easily accessible from the Bestiary Sheet—particularly in the case of K/S Areas, one sometimes needs to go through the Persona Sheet to get to the Config Tab to set up which K/S Areas show up on the Bestiary Sheet. It’s relatively rare for a monster of some sort to have K/Ss, and it’s only likely something one will have to do figure out once, so it shouldn’t be too much of a burden, but I’m open to have a shortcut button on the Bestiary Sheet that goes directly to the Personal Sheet Config Tab.

Back to the details of the Bestiary Sheet, the Dazed (manual)? checkbox performs the same function as it does on the Persona Sheet. The name, identifier, and habitat are pure flavor and are not used by the Roll20 systems. Size, on the other hand, is used to modify the PMSpd and PNSpd into PMSpd(eff) and PNSpd(eff), as both of those stats are always relative to the Size of their possessor. For instance, a Large creature, at 2x human size, with a PMSpd of 40, would have an effective PMSpd of (actual) PMSpd/Size = 20, and initiative rolls or combat BAC or anything else that depends on physical speeds would use PMSpd (eff) or PMSpd (eff). The second size field provides more detail about a creature’s size beyond just comparison to human.

# Appear is a roll which will provide the number of the critters likely to be present, and Alt. # Appear. is the much larger number of such creatures found in a community, if appropriate.

Movement modes and types are available with the Add+ and Modify buttons, so any number of Movement rates (in yds/BT) can be listed or rearranged. There’s a text field available for any notes about movement… for instance, how long a given Movement Mode can be sustained before a switch to other movement is required.

Init Mods are any modifiers to Initiative or Surprise which are not Human Standard.

Next comes Outstanding K/S Areas. Most creatures won’t have these, but sentient ones sometimes will, as well as major Personas. If they have not been cut and pasted in with the Import New button, you will have to go to the main Persona Sheet and the Config Tab to turn some of these on. Once each desired K/S area is checkboxed on the Config Tab, it will show up on the Bestiary view. Things look a little different, here, though. First of all, there is a field for each K/S Area called Range; it provides a range of STEEP amounts and can contain either 1) a single value which will be copied over directly, 2) a die roll range of values like 3d10+15, or 3) the identifier of another K/S Area from which the value will be duplicated. This is of use when a sentient beast has Dweomercraeft capabilities, for instance, and to simplify matters you just want Magick to equal Dweomercraeft General. Clicking on the “Vary” box between the Range and STEEP fields will allow JMs to vary the stats of the K/S, copying over any single values, rolling dice for ranges, or copying from the alternative K/S area, and clicking on the button with the name of the K/S itself, or any of the Sub-Areas accessed by the checkbox located there, will execute a K/S skill check against that K/S Area or Sub-Area marked on the button. Sub-Areas have only one checkbox in this view: Has, to indicate whether the sub-Area is possessed; for the purposes of this sheet, specialization anfd proofing are a non-starter.

After the K/S Areas, if any, is a listing of Joss (along with another Vary button to allow for random tweaking of the number of JFs. Next comes Dodging Factor and Avoidance, computed from basic stats. Note that if Mental TRAIT is marked “cunning”, there will be no Mental Avoidance.

Invulnerabilities and Susceptibilities are where the various items of that ilk are to be found, and extra Sensory Abilities are listed as well.

Average Armor is next, followed by any Quirks, but then we proceed to a checkbox whick can be opened to give the critter or Persona some amount of Heka.

As for Attacks, the next item on the list, a few things must be kept in mind that are different from the Persona View. For one thing, it is permitted to have the Sub-Area field of a weapon filled with No Associated K/S Areas, so it will compute BAC purely from the WeaponMiscBAC parameter. That said, if the critter or Persona is wielding a weapon and has STEEP in the appropriate Sub-Area, the standard weapon fields should be filled out properly; the BAC will be computed peoperly. Note that there are some new forms of damage available in the Bestiary Sheet that are not present in the Persona Sheet.

Powers are handled precisely as they are in the Persona Sheet, though of course critters are likely to have more of them than Personas.

Next are the standard ATTRIBUTE/CATEGORY/TRAIT stats; the only things of note here are: 1) a checkbox to designate Mental TRAIT scores as “cunning”, thereby affecting the use of that TRAIT for some of the other rolls and 2) the presence of PMPow (eff) and PNPow (eff), computed from PMPow and PNPow divided by Size. There also exists a Typical Variation list with guidelines on how to modify the stats for variety in critters, but they have to be applied manually.

Finally is a summary of Armor Scheme. Unlike the Persona Sheet, the Armor section of the Bestiary Sheet is focused on the raw numbers, not attempting to justify them with specific items if armor.

We round out the rest of the Bestiary Sheet with Commentary and Description, where all other descriptive details should be found.

And that’s it. There will be more forthcoming in the next few weeks, but for now, that’s the documentation.

Links:

The Import Templates are available in an accessory folder with the Character Sheet. The algorithm definitely works with Excel 2016 or later, but I’m not sure the Entire Character and Critter imports work with versions earlier than Excel 2016. Better instructions are forthcoming, but here is the link to the Template:

<https://github.com/Roll20/roll20-character-sheets/blob/master/DangerousJourneys-Mythus/accessories/roll20_import_generator2.xlsx>

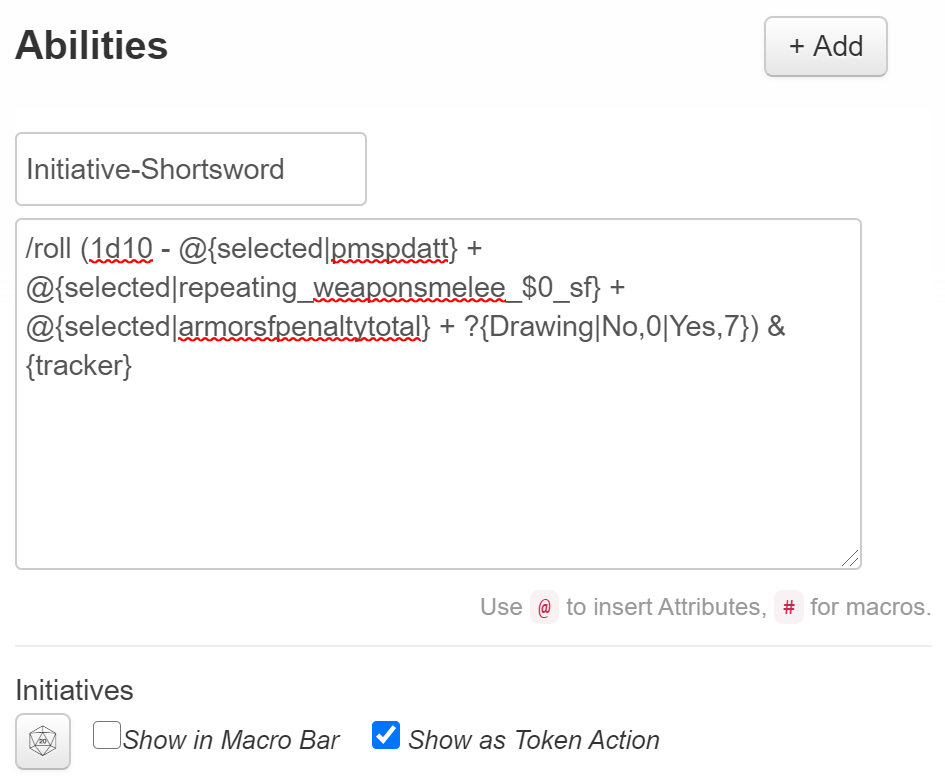
**Roll Template Usage:**

**Inititative:**

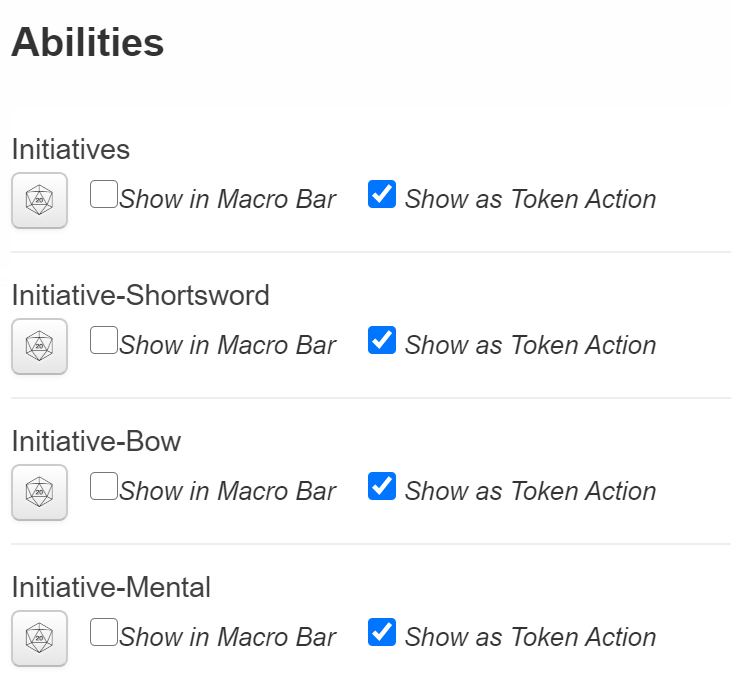
/roll (1d10 - @{selected|pmspdatt} + @{selected|repeating\_weaponsmelee\_$0\_sf} + @{selected|armorsfpenaltytotal} + ?{Drawing|No,0|Yes,7}) &{tracker}

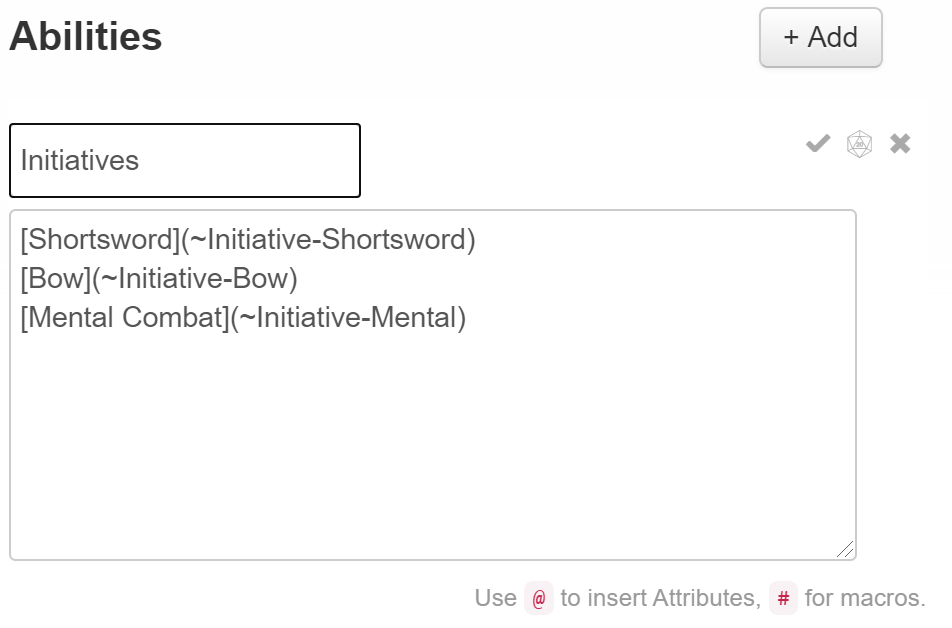
Rolls initiative dice and adds the speed factor of Melee Weapon #0 for the character whose icon is selected, adds the speed factor penalty from the character’s armor, and queries as to whether the weapon was being drawn or not—if it is, the adjustment is made to the initiative roll. The result is applied to the Initiative Tracker.

You can place it as an Ability, and then it will appear as a button when you select your character’s icon.

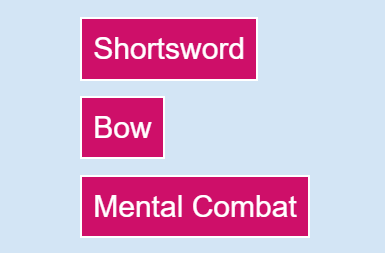


You can have multiple initiatives selectable with one button press:

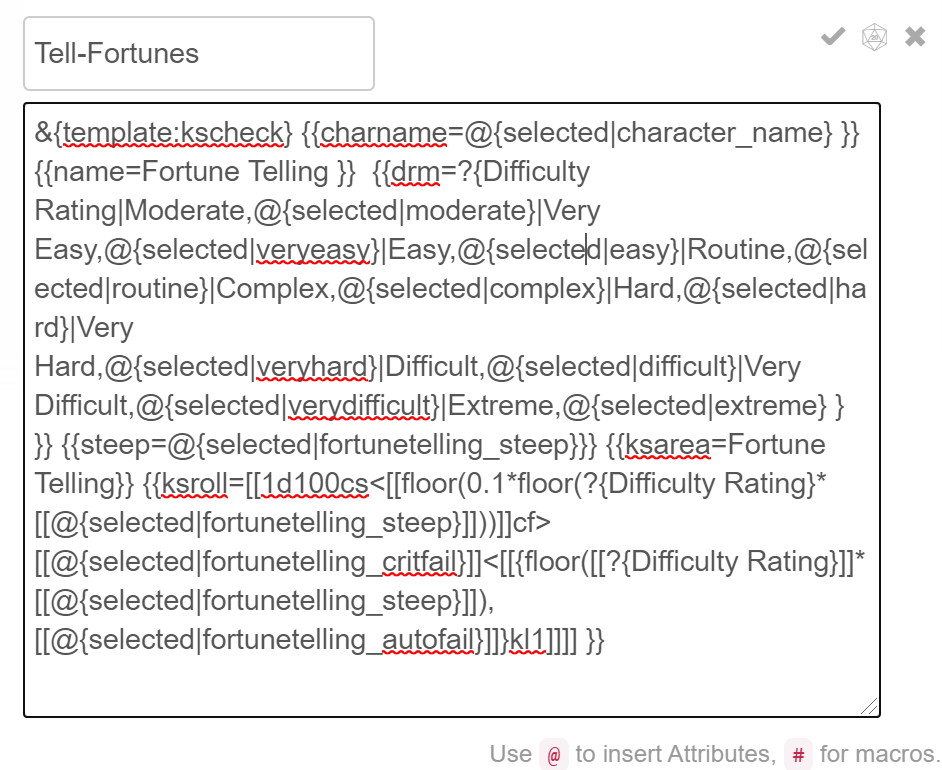




The buttons show up in the chat window and allow selection of various different acts, each of which will show up in the tracker if clicked. Each button is in brackets and each macro name is preceded by a tilde (~) and you can place as many as you like.



**The K/S check template** allows selection of a difficulty rating and rolls against the selected parameter:



It uses the kscheck template, which has parameters built in to specify:

{{charname}}: character name displayed in the header

{{name}}: name of the check being run displayed in the header

{{ksarea}}: name of the K/S area being checked; often but not always redundant with {{name}}

{{steep}}: parameter which is being rolled against, generally the STEEP value of the K/S Area or Sub-Area

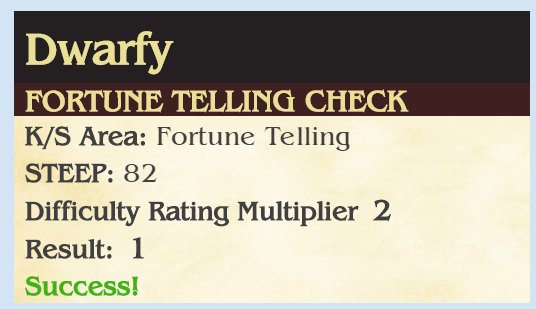
{{msg}}: A message, probably about the difficulty rating

{{dr}}: Difficulty Rating (words)

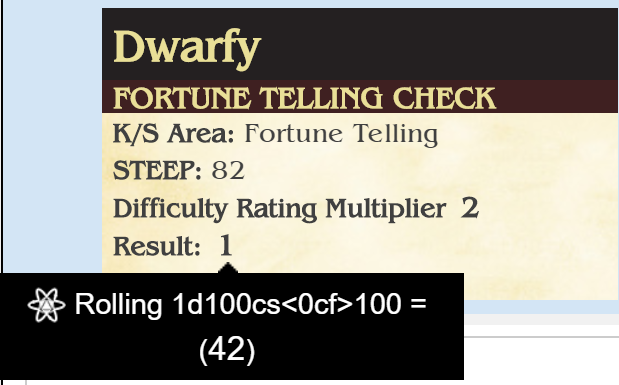
{{drm}}: multiplier associated with the difficulty rating

{{ksroll}}: the actual roll being made

When executed, it generates a message to the chat window that looks like this:



If you want to look at the underlying data behind the roll, hover on the Result value (0 or 1) and it will show up in a bubble:



Let’s go through it step by step…

First… for any of these parameters in {{ … }} you could specify a hardcoded value. In our case, to make the code more portable (so it can be modified easily copied to any other character with minimal effort) we have used variables. For instance, any time you see @{*something*}, it is referring to an attribute in the character sheet. Not all attributes that exist are visible on the sheet, but most can be revealed by hovering over a text box on the sheet. At any rate, if you are clicking a button or template actually located on the sheet, the name of the attribute is enough to get its value into the template.

@{astronomy\_steep}

If, however, you’re typing or pasting the template into the chat box, you have to specify which character’s values to use. You can do this either directly with the name of your character:

@{Dwarfy|astronomy\_steep}

Or indirectly:

@{selected|astronomy\_steep}

@{target|astronomy\_steep}

Using *selected* tells the template to use the stats of whatever character icon is currently selected on the tabletop when the template is executed (either by pressing a button or by typing the template into Chat), while using *target* prompts the user to click on the icon to use while the template is already executing.

You can also allow the user to specify a parameter at execution time by using ?{*prompt*|*value*…} Prompt is the question asked of the user during execution, and value is the possible values allowed, with the first specified as the default value. A second occurrence of the prompt syntax, which has the same Prompt name but no available values, will carry over the value selected earlier.

Let’s examine the entire command in detail:

**&{template:kscheck}** the name of the template we’re using. kscheck is for K/S Checks, attack is for attacks, parry if for parrying, kscontest is a contest between two opponent parameters (usually but not always K/S Area vs K/S Area), power displays the details of a persona or creature’s Power, and casting lists out the name and details of the casting it references.

**{{charname=@{selected|character\_name} }}**

The name of the selected persona will be displayed in the header.

**{{name=Fortune Telling }}** the name of the check being done

**{{dr=Complex}}** the name of the difficulty rating (optional, not used in this example because we want the user to be able to select one instead)

**{{drm=[[?{Difficulty Rating|Moderate,@{selected|moderate}|Very Easy,@{selected|veryeasy}|Easy,@{selected|easy}|Routine,@{selected|routine}|Complex,@{selected|complex}|Hard,@{selected|hard}|Very Hard,@{selected|veryhard}|Difficult,@{selected|difficult}|Very Difficult,@{selected|verydifficult}|Extreme,@{selected|extreme} } ]]}}** The multiplier applied to the roll due to the difficulty rating. Note, this could be as simple as just typing in a value (“2”) or can be a parameter. In the Mythus character sheet, there is a hidden attribute for every DR, named appropriately, which holds the value of the various multipliers. When a persona is Dazed, those multiplier values are modified to 2 DRs higher than normal, per the rules; in normal situations, Hard = 1, but when Dazed it is 0.25. The sheet will do this automagically so you can just specify the attribute instead of trying to compute it on the fly. If this was always supposed to be the same DR, we could hardcode it as, for instance, **@{hard}** or **@{veryeasy}**, but instead we’ve prompted the user for a DR (?{Difficulty Rating?}) and will run with the selected DR as decided on by the user (presumably under JM guidance.).Note the use of [[ ]] around the value. Without the brackets, that query and many others will return the equation referred to, not its end result.

**{{steep=@{selected|fortunetelling\_steep}}}** the name of the attribute used to make the STEEP K/S check.

**{{ksarea=Fortune Telling}}** name of the ksarea being checked

**{{ksroll=[[1d100cs<[[floor(0.1\*floor(?{Difficulty Rating}\*[[@{selected|fortunetelling\_steep}]]))]]cf>[[@{selected|fortunetelling\_critfail}]]<[[{floor([[?{Difficulty Rating}]]\*[[@{selected|fortunetelling\_steep}]]),[[@{selected|fortunetelling\_autofail}]]}kl1]]]] }}**

Rolls are complex, and this is not the right place to get into them, but suffice it to say most checks are of the form

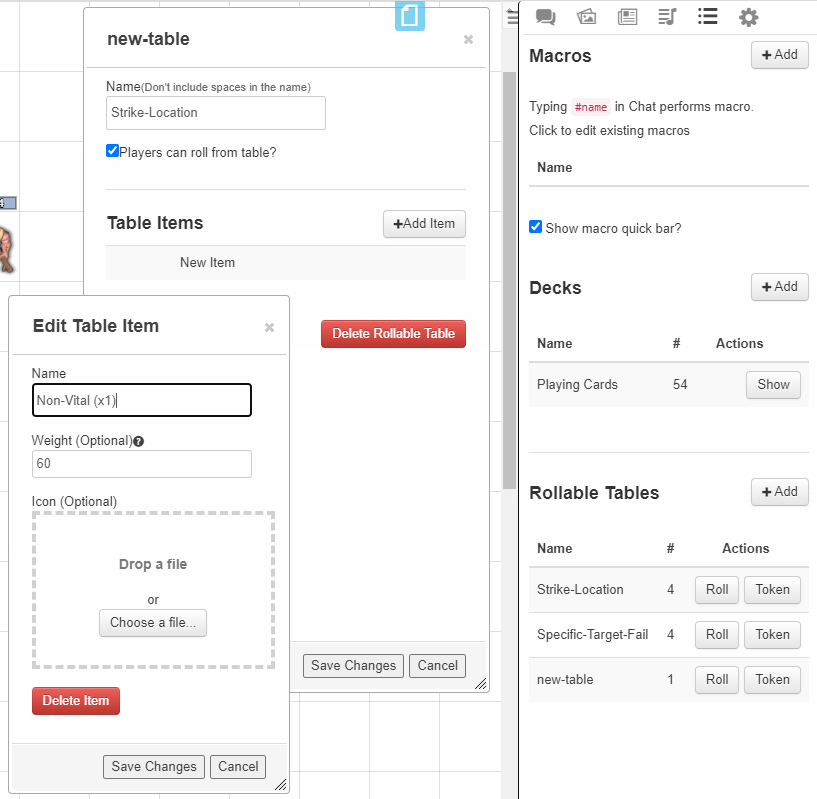
[[dice to roll]]cs<[critical success value]cf>[critical failure value]<[value to “beat” for a Success]

In this case, we are computing a number of these parameters or otherwise reading them from attributes. Critical Success ends up being 0.1\* the difficulty rating \* STEEP; whereas the critical failure value is drawn directly from a parameter present on the sheet for all K/S Areas [ksarea]\_critfail or more specifically **fortunetelling\_critfail**. Finally, the success criteria is defined as DR\*STEEP. Of course, automatic failure is slightly impacted by STEEP, but the sheet recomputes the \_autofail and \_critfail values whenever STEEP changes for any reason; it is for this reason that the last bit after the comma is present: the \_autofail parameter followed by **kl1**. This should be read as “Failure is when die roll is higher than DR\*STEEP or when the die roll exceeds the \_autofail attribute”. (kl1 means look at both the DR\*STEEP and the autofail value and “keep” the lower of the values as the limit). Note: “floor” is a math function which essentially means “round down to nearest integer”.

With that, the kscheck template is complete, and the dice are rolled, and the template indicates Success, Special Success, Failure, or Special Failure. Because this uses “selected” instead of a character name, you could cut a paste this template into any other character sheet and it would work just as well. Moreover, because there’s nothing special about how this computation is done, you could replace “fortunetelling” with the attribute name of any other K/S Area or Sub-Area and it will operate exactly the same, just with the STEEP, critical fail, and autofail parameters of that K/S area instead. Later we’ll look at this in context of the K/S to K/S contests, but there we will use a slightly different template.

The **Attack** template is slightly different but follows the same principles.

Before you start the Attack template, know that **the default buttons assume you have set up a Strike Location table**. I’m trying to figure out a way to easily distribute it, but for now, a brief aside on how to create it. Go to the upper right of the chat window, where you’ll see the Menu icon. Click on it to reveal a menu which allows Macros, card Decks, and Rollable Tables. Click +Add next to the Rollable Tables entry and a new table entry will appear. Click on it and an interactive table creation will occur:



Name the table “Strike-Location”—**be exact in that naming** as that’s what the sheet’s buttons use as a reference. In the Table Item dialog, click the New Item line and you can create a table entry. You will want to do this four times, once for each strike location possibility. You’ll want to fill in the Names and Weights as

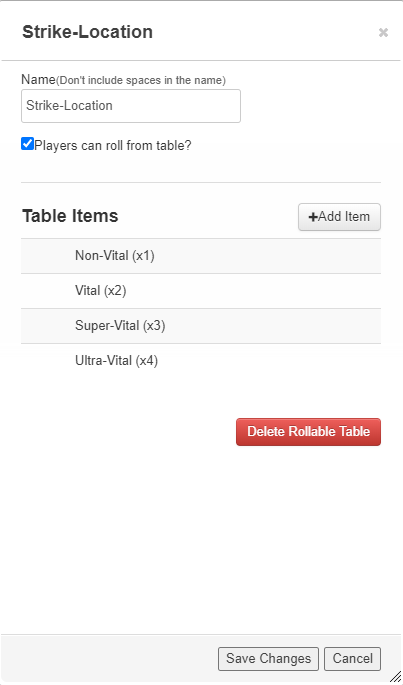
Non-Vital (x1) 60

Vital (x2) 15

Super-Vital (x3) 15

Ultra-Vital (x4) 10

Hit the Save Changes button after each entry, click the “Players can roll from table? Checkbox, and you will have a table that looks like



To roll on this table you can either click the Roll button next to its entry in the chat window



Or type into the Chat window:

/roll 1t[Strike-Location]

The result will be just as if you’d rolled for Strike Location on percentile dice (the weights above are 60+15+15+10 = 100, so the probabilities are the same as d100).

(you can provide alternative tables for different situations… for instance, if you have the Weapons Special Skill Sub-Area of Specific Target, failing your roll against that K/S allows your attack to proceed but with a +20 on the Strike Location table, with numbers over 100 indicating a miss. We can simulate that by creating a Specific-Target-Fail table with new, different weights to it:

Miss 20

Non-Vital (x1) 60

Vital (x2) 15

Super-Vital (x3) 5

This allows you to use that alternative table when there’s a failure against Specific Target. I’ll show this later.

With that out of the way, let’s proceed to look at the variables available in the Attack roll template.

{{charname}} is the name of the character making the attack

{{name}} is the name of the weapon used

{{ksarea}} is an optional parameter where you can name the K/S area used for the attack

{{bac}} is the Base Attack chance for the attack

{{msg}} is an informational message, used in some cases to remind the user why some of the parameters are as they are

{{dodge}} The Dodge Factor of the target of the attack

{{range}} The range modifier of a ranged attack

{{mods}} any other modifiers applied to the Final Attack Chance (FAC)

{{dr}} the difficulty rating (optional; for most attacks it will be Hard)

{{drm}} the actual multiplier number based on the DR (generally 1 for Hard)

{{ksroll}} an optional parameter specifying any roll which might be rolled in conjunction with the attack which may affect it; for instance, a Specific Target roll to determine if the persona can successfully choose her Strike Location.

{{consequence}} what the consequence of the failure of the {{ksroll}} would be… in the Strike Location example above, the consequence would be a roll against the Specific-Target-Fail table

{{rng\_attack}} a roll for a ranged attack

{{mel\_attack}} a roll for a melee attack

{{parry}} message (or possible button!) which appears on a successful attack allowing a parry

{{damage\_dice}} the dice for the damage “1d6+3”, etc. in text form (informational)

{{damage}} same as {{damage\_dice}} above, but the computed value rolled, “[[1d6+3]]” to use the above example; you can append damage type if you like as well.

{{notes}} any notes about the attack or weapon used which could be of use in combat, including whether it can be used to dismount an opponent, how many armor factors it ignores, etc.

Now, let’s see how a standard attack is put together. If you click on a weapon’s BAC button, the chat window gets:



The parry button we will cover later. Suffice it to say that the defender can click that button when she is attacked and attempt a parry… or not.

That’s a standard attack—it queries the user for any modifiers to FAC (for position, movement, etc., per the Core rules)… and here’s how it’s put together:

**&{template:attack}** name of the template used

**{{charname=@{character\_name}}}** character name, placed in the header

**{{name=@{weaponname} }}** the name of the weapon as specified in the character sheet’s weapon list

**{{bac=@{weaponbac} }}** the BAC from that weapon

**{{msg=@{dazemessage} }}** a special attribute, invisible to the user, which indicates when the persona is Dazed and informs/reminds them that things have been modified.

**{{drm=1 }}** default value of Hard for most attacks

**{{mel\_attack=[[1d100cs<[[floor(0.1\*floor(1\*([[@{weaponbac}]]-[[@{target|dodge}]]+[[?{Modifiers to Final Attack Chance?|0}]])))]]cf>[[@{weaponcritfail}]]<[[{floor([[1]]\*([[@{weaponbac}]]-[[@{target|dodge}]]+[[?{Modifiers to Final Attack Chance?}]])),[[@{weaponautofail}]]}kl1]]]] }}**

This should look familiar, as it’s almost identical to the ksroll we did above. A dialog box pops up when the attack proceeds asking the user to identify the icon of the target of the attack so it can figure out the target’s dodge factor, if any, and then inquires if there are situational modifiers to the FAC, defaulting to 0 if the user just hits Enter without changing anything. It sets up the critical fail and autofail values. (NOTE: I do not currently have the capability to modify autofail and critfail for attacks per the rules for FAC higher than 100… Based on certain limitations, I do not know if that will ever be possible. Keep this in mind in cases where your FAC is higher than 100 for whatever reason. You may have to enter them manually in those cases, as you will have to modify DR when the JM has decreed a difference from the default Hard value.)

**{{damage=[[@{weapondmgtot}]] @{weapontype} Damage multiplied by strike location: [[1t[Strike-Location]]]}}** Here’s where the Strike Location table comes into play. We specify the damage roll which should be made (weapondmgtot is the attribute on the character sheet for the chosen weapon) and the type of damage (piercing, cutting, etc.) using @{weapontype}, the attribute on the character sheet for the chosen weapon’s type of damage), and you can see we have placed a reference to the Strike Location table such that when the roll template is executed it reports out what to multiply the damage by according to the Strike Location result. Again, limitations preclude us from computing the damage directly, but I am hoping that since x1, x2, x3, and x4 are relatively simple math problems this will not cause too much distress.

**{{damage\_dice=@{weapondmgtot}}}** This is a duplicate of the damage roll above, but it doesn’t calculate, just reports back what the damage dice are for that weapon for informational purposes. The roll template is set up such that a Special Success will tell you to maximize the damage from that die roll, though, again, suckily, we can’t actually do that computation without some sacrifices I don’t want to make (I could do it so Pro users or people whose JM is a Pro user would get the computation done automatically, but I am trying very hard to make this sheet usable by everyone, not just Pro users).

**{{notes=@{weaponspecial}}}** The weaponspecial parameter is the notes field of every weapon, and this will report those notes out during the attack. You could conceivably even put another roll in here: for instance, if the weapon is capable of dismounting an opponent with a regular attack roll at Difficult, you could do

{{notes=Can dismount a mounted foe; Dismount Roll at Difficult: [[1d100cs<[[floor(0.1\*floor(0.5\*([[@{weaponbac}]]+[[?{Modifiers to Final Attack Chance?|0}]])))]]cf>[[@{weaponcritfail}]]<[[{floor([[0.5]]\*([[@{weaponbac}]]+[[?{Modifiers to Final Attack Chance?}]])),[[@{weaponautofail}]]}kl1]]]] }}

and it will automatically make the Dismount roll during every attack; if you’re not trying to dismount an opponent at that time, just ignore the roll and its results.

**{{mods=?{Modifiers to Final Attack Chance?}}}** This is already embedded in the computation, but it’s placed here informationally to let people know which modifiers were applied to the roll.

**{{dodge=@{target|dodge}}}** This is a repeat of what’s in the roll for informational purposes, to let the user know what the target’s dodge factor was when it was applied to the roll.

**{{parry=[parry](~target|parry)}}** This embeds a button labeled “parry” in the chat window as part of the attack *if* the attack succeeds. It is in place to allow the user who presses it (presumably the player whose persona is the target of the attack) to automatically parry… if that persona has an ability called “parry” (see below)… without having to call up their own macro or open a character sheet, etc.

That covers the standard attack as delivered by the BAC button next to each weapon on the sheet. There are minor improvements *if you re-create the attack as an Ability* on the Attributes and Abilities tab—the levels of indirection required are too much for Roll20’s fragile little mind when calling directly from the character sheet, or I would have built it in—by replacing the parry parameter shown above with

**{{parry=[parry](!&#13;&#37;&#123;@{target|character\_name}&#124;parry&#125;)}}**

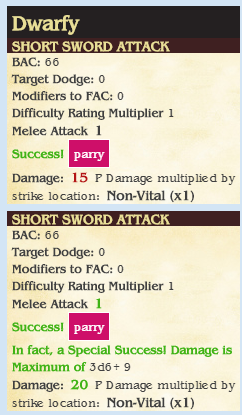
the parry button, when it appears, will already have the character’s name embedded in it, which means you don’t have to select the target icon again during the parry.

Of course, if you are putting the attack in the Abilities tab (which is a terrific idea) you will have to specify your weapon since you’re not pushing a button right next to it on the sheet. You will do that by noting the order of your weapon on the sheet, starting from 0, i.e. the first melee weapon listed is a 0, the second weapon listed is 1, etc. Now, for all the weapon parameters, you need to tell Roll20 which weapon you’re referring to with by prefixing all weapons with **repeating\_weaponsmelee\_$id\_** or **repeating\_weaponsmissile\_$id\_** , where “id” is the number of the weapon. So for the first melee weapon on your list, the new Attack macro will look like:

&{template:attack} {{charname=@{character\_name}}} {{name=@{**repeating\_weaponsmelee\_$0\_**weaponname} }} {{bac=@{**repeating\_weaponsmelee\_$0\_**weaponbac} }} {{msg=@{dazemessage} }} {{drm=1 }} {{mel\_attack=[[1d100cs<[[floor(0.1\*floor(1\*([[@{**repeating\_weaponsmelee\_$0\_**weaponbac}]]-[[@{target|dodge}]]+[[?{Modifiers to Final Attack Chance?|0}]])))]]cf>[[@{**repeating\_weaponsmelee\_$0\_**weaponcritfail}]]<[[{floor([[1]]\*([[@{**repeating\_weaponsmelee\_$0\_**weaponbac}]]-[[@{target|dodge}]]+[[?{Modifiers to Final Attack Chance?}]])),[[@{**repeating\_weaponsmelee\_$0\_**weaponautofail}]]}kl1]]]] }} {{damage=[[@{**repeating\_weaponsmelee\_$0\_**weapondmgtot}]] @{**repeating\_weaponsmelee\_$0\_**weapontype} Damage multiplied by strike location: [[1t[Strike-Location]]]}} {{damage\_dice=@{**repeating\_weaponsmelee\_$0\_**weapondmgtot}}} {{notes=@{**repeating\_weaponsmelee\_$0\_**weaponspecial}}} {{mods=?{Modifiers to Final Attack Chance?}}} {{dodge=@{target|dodge}}} {{parry**=[parry](!&#13;&#37;&#123;@{target|character\_name}&#124;parry&#125;)}}**

You can name that ability Attack-ShortSword, click Show As Token Action, and then whenever you select that character’s icon, Attack-ShortSword will appear as a button on the tabletop and clicking on it will call that Ability, which is an attack by short sword by the selected character which waits for you to select a target icon, then uses that target’s dodge factor to decide whether a hit occurs or not, and then submits a button for the defender to use to parry. A lot of hard work, but pretty slick in the big scheme of things.

Since I know that I have two attacks per CT with that particular weapon, I could in fact paste the template twice into the same Ability and it will execute twice each time I push the button.



I could even make it so the first attack and second attack are permitted to select separate targets by replacing *{target* with *{target|target1* in the first template and *{target|target2* in the second template. The possibilities, while not limitless, are pretty cool.

Another option uses the aforementioned Specific-Target-Fail table (remember creating that?), placing a kscheck roll in the optional {{ksroll}} field of the attack template along with using the {{consequence}} field to detail what happens on a failure:

**{{ksroll=[[1d100cs<[[floor(0.1\*floor([[?{Strike Location Attempted|Non-Vital,3|Vital,2|Super-Vital,1|Ultra-Vital,0.5}]]\*[[@{ specifictarget\_steep}]]))]]cf>[[@{specifictarget\_critfail}]]<[[{floor([[?{Strike Location Attempted}]]\*[[@{specifictarget\_steep}]]),[[@{specifictarget\_autofail}]]}kl1]]]] }} {{consequence=Unfortunately, the hit location is overridden with [[1t[Specific-Target-Fail]]]}}**

When an attack is made, the user will be queried as to what area they are trying to hit, DR is computed according to the rules, and if the attack hits, but the Specific Target roll is failed, you will be informed of what the actual target was. It can’t overwrite what’s written in the successful attack field—that’s just not how Roll20 rolls—but it can provide you with all the information you need to decide how to actually apply the damage.

The **Parry** roll template is designed to automate the parrying rolls. It could still use a little work, as it doesn’t take into account weapon composition yet, but it’s usable for the basics. It can be used independently or embedded as a button in the attack template as already discussed. Let’s start with the basics, the parameters:

{{charname}} is the name of the character making the attack

{{name}} is the name of the weapon or shield used to parry

{{bac}} is the Base Attack chance for that weapon/shield

{{msg}} is typically used to remind you whether you’re Dazed or not so you know why your DR is worse than usual

{{dr}} the difficulty rating (optional; usually this will be chosen dynamically by the user)

{{drm}} the actual multiplier number based on the DR (dependent on the type of weapon being parried)

{{parry}} the roll equation used to attempt the parry

{{counter}} the equation used to determine if the parried weapon is damaged in the parry

{{damage\_dice}} the dice for the damage which might be inflicted on the opposing weapon “1d6+3”, etc. in text form (informational)

{{damage}} same as {{damage\_dice}} above, but the computed value rolled, “[[1d6+3]]” to use the above example.

{{notes}} any notes about the attack or weapon used which could be of use in parrying.

Parry templates are best used as an Ability, but given the nature of how parrying is often used we’ll want to make sure we keep our character name in the reference to any variables since we can be fairly certain we will be using it as a button in someone else’s chat and therefore Roll20 has to be told explicitly who the parry is about. So here’s an example I called “Parry-Short-Sword”:



Piece by piece:

**&{template:parry} {{charname=@{Dwarfy|character\_name}}} {{name=@{Dwarfy|repeating\_weaponsmelee\_$0\_weaponname} }} {{bac=@{Dwarfy|repeating\_weaponsmelee\_$0\_weaponbac} }} {{msg=@{Dwarfy|dazemessage} }}**

We specify the name of the template, and explicitly call out our character name before each variable from the sheet.

**{{drm=?{Weapon Parried|Fist/Kick,[[@{Dwarfy|easy}]]|Large thrown weapon,[[@{Dwarfy|moderate}]]|Hand weapon,[[@{Dwarfy|hard}]]|Medium thrown weapon,[[@{Dwarfy|difficult}]]|Small thrown weapon,[[@{Dwarfy|verydifficult}]]} }}**

We query the user as to what kind of weapon is being parried and assign a difficulty rating accordingly. That one should work for all true weapons; for parrying using fists/kicks, you would eliminate the “easy” rating and start at “moderate”, even for parrying other fists/kicks. For shields you would use:

**{{drm=?{Weapon Parried|Fist/Kick or large/medium thrown weapon,[[@{Dwarfy|easy}]]|Hand weapon,[[@{Dwarfy|moderate}]]|Small thrown weapon,[[@{Dwarfy|hard}]]|Arrows/bolts/sling stones/etc.,[[@{Dwarfy|difficult}]]} }}**

Next is the actual roll:

**{{parry=[[1d100cs<[[floor(0.1\*floor([[?{Weapon Parried}]]\*([[@{Dwarfy|repeating\_weaponsmelee\_$0\_weaponbac}]])))]]cf>[[@{Dwarfy|repeating\_weaponsmelee\_$0\_weaponcritfail}]]<[[{floor([[?{Weapon Parried}]]\*([[@{Dwarfy|repeating\_weaponsmelee\_$0\_weaponbac}]])),[[@{Dwarfy|repeating\_weaponsmelee\_$0\_weaponautofail}]]}kl1]]]] }}**

The parry roll looks just like an attack roll with the weapon in question except it uses the DR appropriate to the kind of weapon it’s parrying.

**{{damage\_dice=@{Dwarfy|repeating\_weaponsmelee\_$0\_weapondmgtot}}} {{damage=[[@{Dwarfy|repeating\_weaponsmelee\_$0\_weapondmgtot}]] }} {{notes=@{Dwarfy|repeating\_weaponsmelee\_$0\_weaponspecial}}}**

Again, other stuff that looks just like an attack roll. The damage is used to compute how much damage is done to the opposing weapon if needed.

**{{counter=Weapon vs Weapon Result: [[1d100]] [opposing weapon](~target|parry) }}**

This details what happens to the weapon when the attack goes through, and creates a button for the attacker to roll his own weapon damage parry roll against the defender’s weapon. In this case, there’s a roll of 1d100 which needs to be compared against the opposing weapon’s durability number… alas, that will be manual, generally speaking. If you knew what that weapon was, you could do something like

**{{counter=Weapon vs Weapon Result: [[1d100<[[@{target|repeating\_weaponsmelee\_$0\_weapondurhard}]] [opposing weapon](~target|parry) }}**

Here we have guessed that we will be parrying the first weapon on the opponent’s list (not a terrible assumption, but certainly not always accurate, so I default to leaving it alone instead of guessing), and grabbing the weapondurhard, trying to roll under it to score damage. The opposing weapon button will allow the attacker the same opportunity.

Now… when attacked, what if you’re not holding your short sword? What if you’re parrying with a shield, or this time you’re using your longsword? You can make a parry template for each of these individually:

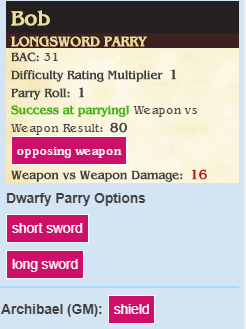


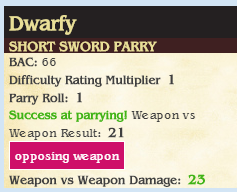
Here’s where the magic comes in… the attack roll template creates a button which calls the ability “parry”. You should make an ability “parry” which allows you to choose which weapon or shield you’re parrying with:



Now, when the attacker hits and a button shows up for you to parry and you press it, your own parry command pops up three buttons for you to choose from, and you can click which one you’re parrying with and it will execute. The entire exchange can look like this:







Dwarfy hit Bob twice. Bob’s first parry with his longsword was unsuccessful, so he should take the damage from the original attack roll and ignore the Weapon vs Weapon damage reported in his parry. However, his second parry was a success, though he only rolled 80 when trying to do 16 points of damage to Dwarfy’s short sword. Dwarfy’s return roll would have done 23 points to Bob’s long sword, but a 21 Weapon vs Weapon result is probably still too high to do damage… durability numbers are typically in the single digits unless the weapon is below average or wood or both.

The parry roll template might be improved with more capability, eventually, taking weapon composition into account, but for now that’s what we have.

The **K/S vs K/S contest** template is for times when we must roll one persona’s stat against another’s—either the same stat or a different stat… for instance opposed Riding checks when racing one another on horseback, or when trying to create a Link during Mental combat by performing a contest between one stat (Dweomercræft STEEP) and, say, the opponent’s MR Category.

The parameters are as follows:

{{charname}} is the name of the character executing the operation

{{name}} is the name of the contest, “Mental Attack Link” or “Riding”

{{instigator}} the character who is initiating the roll, or if it’s mutual, just the first character in the list

{{respondent}} the character who is responding to the roll, or just the second character in the list

{{initial}} the name of the stat the instigator is rolling against in the contest

{{response}} the name of the stat the respondent is rolling against in the contest

{{initialvalue}} the actual value of the stat being rolled against by the instigator

{{responsevalue}} the actual value of the stat being rolled against by the respondent

{{instdelta}} the instigator’s stat minus the respondent’s stat, used to compute DR

{{respdelta}} the respondent’s stat minus the instigator’s stat, used to compute DR

{{instbutton}} a button created for the instigator to roll the K/S vs K/S check

{{respbutton}} a button created for the respondent to roll the K/S vs K/S check

{{notes}} any notes about what the results might mean

By the nature of having to perform a check across multiple characters, the K/S vs K/S template can end up looking rather complicated. At its simplest and least generic, you could write an up an Astronomy contest without buttons and just rely on users to make their own checks:

&{template:kscontest} {{charname=@{character\_name}}} {{name=Astronomy}} {{instigator=@{character\_name}}} {{respondent=@{target|character\_name}}} {{initial=Astronomy}} {{response=Astronomy}} {{initialvalue=@{astronomy\_steep}}} {{responsevalue=@{target|astronomy\_steep}}} {{instdelta=[[[[@{astronomy\_steep}]]-[[@{target|astronomy\_steep}]]]]}} {{respdelta=[[[[@{target|astronomy\_steep}]] - [[@{astronomy\_steep}]]]]}}

But that’s very tailored to a very specific situation and not very useful and relies on folks opening their character sheets to make the rolls, or typing them into the chat window. The uglier version is more interesting:



**&{template:kscontest} {{charname=@{character\_name}}} {{name=Astronomy}} {{instigator=@{character\_name}}} {{respondent=@{target|character\_name}}} {{initial=Astronomy}} {{response=Astronomy}} {{initialvalue=@{astronomy\_steep}}} {{responsevalue=@{target|astronomy\_steep}}}**

Setting up all the values for the two parties.

**{{instdelta=[[[[@{astronomy\_steep}]]-[[@{target|astronomy\_steep}]]]]}} {{respdelta=[[[[@{target|astronomy\_steep}]] - [[@{astronomy\_steep}]]]]}}**

Establishes the parameters for the Difficulty Rating

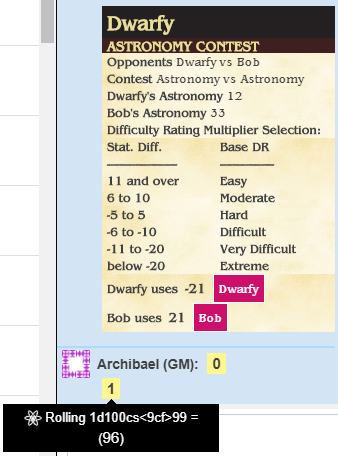
**{{instbutton=[@{character\_name}](!&#13; [[1d100cs<[[floor(0.1&#42;floor([[&#63;{Difficulty Rating1|Easy,[[@{easyshadow}&#93;&#93;|Moderate,[[@{moderateshadow}&#93;&#93;|Hard,[[@{hardshadow}&#93;&#93;|Difficult,[[@{difficultshadow}&#93;&#93;|Very Difficult,[[@{verydifficultshadow}&#93;&#93;|Extreme,[[@{extremeshadow}&#93;&#93;}&#93;&#93; &#42;[[(@{astronomy\_steep}&#41;&#93;&#93;&#41;&#41;&#93;&#93;cf>[[@{astronomy\_critfail}&#93;&#93;<[[{floor([[&#63;{Difficulty Rating1}&#93;&#93;&#42;[[(@{astronomy\_steep}&#41;&#93;&#93;&#41;,[[@{astronomy\_autofail}&#93;&#93;&#125;kl1&#93;&#93;&#93;&#93;)}}**

Complicated. All the &#number; stuff is confusing as heck. Essentially, it’s cryptic ASCII code for various characters like “}” and “@” and other things we would like to see in the macro we map to the buttons on the fly, but which would get interpreted wrongly by the Abilities execution engine if we typed in outright. What this is doing is creating a button in the chat window for the K/S Contest instigator that will enable him to automatically roll his part of the contest and select the DR based on the table and difference stat that the template reports. This saves him from having to open his sheet, find the relevant K/S, and click that button.

**{{respbutton=[@{target|character\_name}](!&#13; [[1d100cs<[[floor(0.1&#42;floor([[&#63;{Difficulty Rating2|Easy,[[@{target|easyshadow}&#93;&#93;|Moderate,[[@{target|moderateshadow}&#93;&#93;|Hard,[[@{target|hardshadow}&#93;&#93;|Difficult,[[@{target|difficultshadow}&#93;&#93;|Very Difficult,[[@{target|verydifficultshadow}&#93;&#93;|Extreme,[[@{target|extremeshadow}&#93;&#93;}&#93;&#93; &#42;[[(@{target|astronomy\_steep}&#41;&#93;&#93;&#41;&#41;&#93;&#93;cf>[[@{target|astronomy\_critfail}&#93;&#93;<[[{floor([[&#63;{Difficulty Rating2}&#93;&#93;&#42;[[(@{target|astronomy\_steep}&#41;&#93;&#93;&#41;,[[@{target|astronomy\_autofail}&#93;&#93;&#125;kl1&#93;&#93;&#93;&#93;)}}**

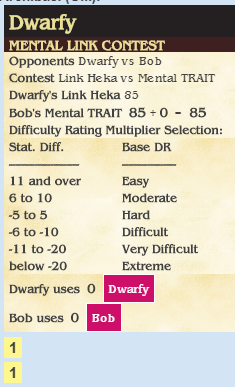
And this is the corresponding button for the respondent; again, so no one has to open a character sheet unneccesarily.

The contest looks like:



Dwarfy’s player presses the button, consults the table, and selects the DR (Extreme, in this case, since he’s got a 21 point STEEP deficit versus his opponent) he should be using from the table; Bob’s player does the same with his own DR (Easy). I don’t love the way the resultant rolls are just presented as numbers 0 or 1 for Failure and Success, but I’m working on it.

Now one could certainly argue that the rarity of an Astronomy contest would not warrant the creation of this as an Ability (“Who can compute Venus’s retrograde motion fastest?”) … and one would be right. Taking care of one-off contests can certainly be better served by just opening the darn character sheets and consulting the K/S vs K/S tables from the Core rules. But what about the persona with *Wound, Mental* that tends to use it quite frequently? Making Mental Links is entirely worth automating (though the exact rules are, of course, under debate… too long a story to discuss here) in this fashion.



That’s all for roll templates for now. I need to add \_critfail and \_autofail parameters for all the attributes, categories, and traits so any K/S rolls against them can use the standard “high STEEP” rules for when a failure occurs, and I may create a special template for Avoidance rolls (though you could probably just use a standard K/S check for those, technically) but unless there are other rolls people are frequently making that don’t work with this set, there aren’t at this point plans to make more embedded in the character sheet. Please feel free to request some, though!

**Changelog:**

**Rev 0.97:**

* Added roll template for Castings; all Casting rolls adjusted to the new template
* Added roll template and button for Powers
* Added roll template for K/S Checks; all K/S rolls and Sub-Area rolls adjusted to new template
* Added Attack roll template and added it to all weapons buttons
* Added Parry roll template
* Added KS Contest template
* Changed to Benguiat font to closely replicate original Mac Mirage font; parchment background
* Cleaned up cosmetics with new styles
* Added headers to Weapons fields
* Reordered Armor details to better match books
* Heka Aperture override field for use when you don’t want to compute it
* Upgrade 1-time script now accommodates rev 0.96 and rev 0.97 changes separately
* Added new Dodging optional rule to Config page