

# Encounterra Homebrew JSON v0.1

Start with a creature and one attack. Build up to reusable traits, spellcasting, and unusual monster abilities.

Beta v0.1

Schema "0.1"

D&D 2024 rules

SRD catalog 5.2.1-encounterra.1

**Beta v0.1 · Keep an exported copy**

The editor and format are still evolving. This guide explains the currently supported authoring forms. Keep a backup of your JSON and use the editor's error messages to correct unsupported fields or missing references.

**On this page**

1. Make your first creature

3. Describe monsters and heroes

5. Choose targets and areas

7. Apply conditions and end them

9. Add limited uses and spell slots

11. Make traits respond to events

13. Reuse and adapt definitions

15. Add temporary bonuses and penalties

17. Model advanced monster abilities

19. Leave an area on the battlefield

21. Ten complete worked examples

2. Understand the package

4. Add actions and spells

6. Give creatures reusable abilities

8. Combine attacks, saves, and effects

10. Calculate values with expressions

12. Customize grants with parameters and choices

14. Equipment, weapons, and masteries

16. Build random abilities

18. Change a creature's form

20. What to expect in Beta

# 1. Make your first creature

This guide is for writing monsters and heroes that Encounterra can simulate. DEF is the definition format used inside a Homebrew JSON package. You describe what a creature can do using data: its statistics, available actions, and explicit effects. You do not write scripts.

Read the early chapters in order, then use the field tables when you need a detail. Small examples build on the Bramble Scout below. The final chapter contains ten complete, more advanced packages: six monsters and four heroes. Each of those packages is printed in full and can also be downloaded.

## A creature and one attack

Save the following complete JSON as `my-first-creature.json`, or download the starter. Import it in the Homebrew editor and choose Bramble Scout when adding a combatant. It has 25 HP, AC 13, walking speed 30 feet, and an attack at +4 that deals 1d6 + 2 piercing damage.

### Your first complete package

```
{
  "format": "encounterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "f916c49a-a9e9-4d3c-87ad-3e7a98088f65",
    "name": "My First Creature", "version": "0.1.0"
  },
  "requires": {
    "ruleset": "dnd-2024", "srdCatalog": "5.2.1-encounterra.1"
  },
  "definitions": [
    {
      "kind": "monster", "id": "bramble-scout", "name": "Bramble Scout",
      "stats": {
        "hitPoints": 25, "armorClass": 13, "speedFeet": 30,
        "size": "small", "creatureType": "plant"
      },
      "actions": ["#thorn-strike"]
    },
    {
      "kind": "action", "id": "thorn-strike", "name": "Thorn Strike",
      "activation": "action",
      "target": { "side": "enemy", "rangeFeet": 5 },
      "resolution": {
        "kind": "attack", "attackBonus": 4,
        "damage": {
          "dice": { "count": 1, "sides": 6 }, "bonus": 2,
          "damageType": "piercing"
        }
      }
    }
  ]
}
```

Read it from the inside out. The `monster` entry describes the creature. Its `actions` list gives it Thorn Strike. The `action` entry says when that attack can be used, who it can target, and what happens. The surrounding package keeps

those entries together in one importable file.

## Try one change at a time

Change `hitPoints` from 25 to 35, or `damage.bonus` from 2 to 3. Import your edited file and inspect the creature and a short combat replay. Changing the name to “Veteran Scout” changes only its label; it does not add damage or abilities.

Use valid JSON: quoted property names, no comments, and no trailing commas. Later snippets labeled “fragment” belong inside the named field, while lists of definitions belong inside `definitions`. Do not paste a second definition with an ID that already exists: edit or replace the original entry.

## 2. Understand the package

Every exchange file is one JSON object. Package identity and requirements are mandatory, even when the package has no definitions yet.

| FIELD                            | TYPE                      | REQUIRED | MEANING                                                                                                                                                                                                                   |
|----------------------------------|---------------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>format</code>              | string constant           | Yes      | <code>encounterra.homebrew</code> . This prevents unrelated JSON from being imported accidentally.                                                                                                                        |
| <code>schemaVersion</code>       | string constant           | Yes      | Use <code>"0.1"</code> for Beta v0.1, including the quotes. Numeric <code>0.1</code> is invalid. Legacy integer <code>1</code> remains readable without rewriting stored snapshots. This is separate from package SemVer. |
| <code>package.id</code>          | UUID string               | Yes      | Permanent package identity. Generate a new UUID when forking someone else's package as an independent package.                                                                                                            |
| <code>package.name</code>        | string, 1–120 UTF-8 bytes | Yes      | Human-readable library and export name. Non-ASCII characters can occupy more than one byte.                                                                                                                               |
| <code>package.version</code>     | SemVer                    | Yes      | <code>major.minor.patch</code> , optionally with valid prerelease/build identifiers.                                                                                                                                      |
| <code>requires.ruleset</code>    | string constant           | Yes      | <code>dnd-2024</code> .                                                                                                                                                                                                   |
| <code>requires.srdCatalog</code> | string constant           | Yes      | <code>5.2.1-encounterra.1</code> . References are resolved against this locked catalog.                                                                                                                                   |
| <code>definitions</code>         | array                     | Yes      | Up to 2,048 typed definitions. IDs must be unique within the package.                                                                                                                                                     |

### Definitions are the parts inside the package

All definitions share `kind`, `id`, and optional `name`, `extends`, inheritance controls, and parameters. IDs use lowercase letters, digits, and hyphens and must begin with a letter or digit.

| FIELD                | TYPE                       | REQUIRED | MEANING                                                                                                                                                                                                            |
|----------------------|----------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>monster</code> | combatant                  | No       | A complete non-player combatant snapshot: stats, actions, rules, resources, choices, and reusable grants.                                                                                                          |
| <code>hero</code>    | combatant                  | No       | A configured level snapshot. Beta v0.1 does not model level-by-level class progression.                                                                                                                            |
| <code>weapon</code>  | executable                 | No       | A reusable weapon profile with native weapon delivery. Public SRD weapons derive attack and damage from the wielder when referenced by its actions.                                                                |
| <code>action</code>  | executable                 | No       | An action-economy entry with targeting, payment, and one resolution tree.                                                                                                                                          |
| <code>spell</code>   | executable                 | No       | A spell resolution independent of its payment mode; supports level and concentration metadata.                                                                                                                     |
| <code>ability</code> | bundle                     | No       | Reusable actions, event rules, resources, choices, and nested grants.                                                                                                                                              |
| <code>feat</code>    | bundle                     | No       | A reusable feat-shaped bundle. Public SRD feat references are discoverable in the catalog.                                                                                                                         |
| <code>mastery</code> | bundle / weapon capability | No       | A reusable bundle, or a weapon-bound capability with mastery: nick, graze, sap, vex, push, slow, topple or cleave. Bind executable capabilities to <code>weapon.mastery</code> rather than granting them globally. |

| FIELD               | TYPE        | REQUIRED | MEANING                                                                         |
|---------------------|-------------|----------|---------------------------------------------------------------------------------|
| <code>effect</code> | rule bundle | No       | A reusable rule/effect definition with optional resources, choices, and grants. |

Inline action and rule objects are allowed inside combatants and bundles. Named top-level definitions are usually easier to reuse and adapt.

### 3. Describe monsters and heroes

A `monster` and a `hero` both describe a ready-to-play combatant. Use `monster` for a creature labeled by challenge rating and `hero` for a configured level. Neither label automatically grants class progression, spells, attacks, or Hit Dice.

For the Scout, changing `size` to `large` changes its space; it does not change its HP or damage. Give a swimming creature `swimSpeedFeet` as well as `speedFeet` if it should move both ways. Write attack bonuses, save bonuses, and damage explicitly rather than expecting an ability score to fill them in.

#### Use finished ability scores

Write the final ability scores in `stats.abilityScores`, with increases from character creation, leveling and feats already included. For example, if training raised Strength from 16 to 18, write `"strength": 18`. Do not add an Ability Score Improvement grant or a separate score-increase entry: DEF describes the finished combatant, not the steps used to build it.

Update any written attack, damage, saving-throw, HP or AC values affected by that change. Literal numbers are not automatically recalculated; ability-based expressions and documented defaults use the stated final scores. If a feat also provides a combat benefit, model that benefit separately when supported. Grappler is documentation only in Beta v0.1: its feat benefits and tactical grappling are not simulated. Record its name and an explicit limitation in a descriptive trait, without adding combat mechanics. Baking in its score increase does not grant the feat's other mechanics. Separately granted bonuses, such as Defense's AC bonus, must not also be baked into their base fields.

Likewise, put Skilled's relevant proficiencies directly in the finished `athletics`, `acrobatics`, `stealth` and `passivePerception` fields. These are final bonuses or totals, not lists of skill names. There is no need for a Skilled grant. Other skills and tool proficiencies have no combat rules in this format; describe them in your creature's notes if useful.

| FIELD                              | TYPE                           | REQUIRED | MEANING                                                                                                                                                                                                                                                                   |
|------------------------------------|--------------------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>stats.level</code>           | integer 0–20                   | No       | Heroes require 1–20; monsters also permit zero. This is the owner level used by expressions, not automatic class progression or a CR-to-level conversion.                                                                                                                 |
| <code>stats.challengeRating</code> | string                         | No       | Display and catalog CR for monsters.                                                                                                                                                                                                                                      |
| <code>stats.hitPoints</code>       | integer 1–100,000              | No       | Maximum starting hit points.                                                                                                                                                                                                                                              |
| <code>stats.hitDice</code>         | optional { count, size }       | No       | Rest-recovery pool. Count is an integer 0–100; size is 4, 6, 8, 10, 12, or 20. Both fields are required when present. Zero means no Hit Dice. Omitted: heroes use level d8 and monsters use 1d8, preserving saved-package behavior. This does not recalculate maximum HP. |
| <code>stats.armorClass</code>      | integer 1–50                   | No       | Base Armor Class, already including worn armor/shield contributions but excluding separately granted typed AC modifiers. Equipment facts do not recalculate this number.                                                                                                  |
| <code>stats.speedFeet</code>       | integer 0–1,000, multiple of 5 | No       | Walking speed in feet.                                                                                                                                                                                                                                                    |
| <code>stats.flySpeedFeet</code>    | integer 0–1,000, multiple of 5 | No       | Flying speed in feet.                                                                                                                                                                                                                                                     |
| <code>stats.swimSpeedFeet</code>   | integer 0–1,000, multiple of 5 | No       | Swimming speed in feet.                                                                                                                                                                                                                                                   |
|                                    |                                |          |                                                                                                                                                                                                                                                                           |

| FIELD                                                                                                          | TYPE                           | REQUIRED | MEANING                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------|--------------------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>stats.burrowSpeedFeet</code>                                                                             | integer 0–1,000, multiple of 5 | No       | Burrowing speed in feet.                                                                                                                                                     |
| <code>stats.darkvisionFeet</code>                                                                              | integer 0–1,000, multiple of 5 | No       | Native darkvision range.                                                                                                                                                     |
| <code>stats.blindsightFeet</code>                                                                              | integer 0–1,000, multiple of 5 | No       | Blindsight range. Default 0. Uses native special-sense visibility rules; does not imply unrestricted sight through barriers.                                                 |
| <code>stats.initiativeBonus</code>                                                                             | integer –50–50                 | No       | Optional explicit initiative modifier. Omitted: Dexterity modifier. Useful when a creature's initiative differs from its Dexterity.                                          |
| <code>stats.passivePerception</code> / <code>athletics</code> / <code>acrobatics</code> / <code>stealth</code> | integers –50–50                | No       | Optional explicit skill/sense values. Defaults: 10 + Wisdom modifier for passive perception, Strength modifier for Athletics, Dexterity modifier for Acrobatics and Stealth. |
| <code>stats.packTactics</code>                                                                                 | boolean                        | No       | Default false. Grants attack Advantage when an eligible ally is adjacent to the target.                                                                                      |
| <code>stats.exhaustionImmune</code>                                                                            | boolean                        | No       | Default false. Uses the native exhaustion-immunity trait; not an unrelated condition-immunity string.                                                                        |
| <code>stats.difficultyClass</code>                                                                             | integer 0–50                   | No       | Default save DC available to the combatant.                                                                                                                                  |
| <code>stats.spellAttackBonus</code>                                                                            | integer –50–50                 | No       | Default spell attack modifier.                                                                                                                                               |
| <code>stats.size</code>                                                                                        | size enum                      | No       | <code>tiny</code> , <code>small</code> , <code>medium</code> , <code>large</code> , <code>huge</code> , or <code>gargantuan</code> .                                         |
| <code>stats.creatureType</code>                                                                                | creature-type enum             | No       | Used by selectors and creature-type exclusions.                                                                                                                              |
| <code>stats.abilityScores</code>                                                                               | ability map                    | No       | Final Strength through Charisma scores, already including all character-building and feat increases. No separate ASI grant is needed.                                        |
| <code>stats.savingThrows</code>                                                                                | ability map                    | No       | Final saving-throw modifiers, not proficiency flags.                                                                                                                         |
| <code>stats.resistances</code> / <code>immunities</code> / <code>vulnerabilities</code>                        | damage-type arrays             | No       | Damage adjustments applied by the normal combat engine.                                                                                                                      |
| <code>stats.conditionImmunities</code>                                                                         | condition array                | No       | Conditions the combatant cannot receive.                                                                                                                                     |
| <code>stats.magicResistance</code>                                                                             | boolean, default false         | No       | Advantage on saves against spells and other magical effects. Uses native save machinery, not a flat save bonus.                                                              |
| <code>stats.legendaryResistanceUses</code>                                                                     | integer 0–20, default 0        | No       | Finite native save-result replacements. A normal success spends nothing; turn refresh and short rest do not refill the long-rest pool.                                       |
| <code>stats.legendaryActionUses</code>                                                                         | integer 0–20, optional         | No       | Explicit legendary-action point budget, including zero. If omitted, granting a legendary action supplies the normal three-point default. Costs belong on each action.        |
| <code>stats.dimLightResistances</code>                                                                         | damage-type array              | No       | Additional resistances while the placed owner is in actual dim light or darkness. Empty by default; darkvision does not turn off these resistances.                          |
|                                                                                                                |                                |          |                                                                                                                                                                              |

| FIELD                                     | TYPE                   | REQUIRED | MEANING                                                                                                                                                                                   |
|-------------------------------------------|------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>stats.brightLightSensitivity</code> | boolean, default false | No       | Disadvantage on attacks and ability checks, including initiative, while the placed owner is actually in bright light. This is bright-light sensitivity, not a separate sunlight detector. |
| <code>tags</code>                         | array                  | No       | <code>wears_metal</code> is the currently executable equipment tag.                                                                                                                       |
| <code>actions / rules / grants</code>     | arrays                 | No       | Directly written or referenced mechanics belonging to this creature.                                                                                                                      |
| <code>resources / choices</code>          | arrays                 | No       | Pools of limited uses and options chosen when placing this creature.                                                                                                                      |

Defaults are 30 HP, AC 13, speed 30 feet, medium size, monstrosity, level 1, spell attack bonus +2, and DC 10. Ability scores default to 10 and saving-throw modifiers to 0. Special movement and darkvision default to 0; resistance and immunity lists are empty. Author every material stat explicitly.

Ability-map keys are `strength`, `dexterity`, `constitution`, `intelligence`, `wisdom`, and `charisma`. Scores range from 1–30; save modifiers from –50 to 50. Hero levels are 1–20; explicit monster levels may be 0–20 and are separate from challenge rating. Creature types are aberration, beast, celestial, construct, dragon, elemental, fey, fiend, giant, humanoid, monstrosity, ooze, plant, and undead.

## 4. Add actions and spells

An action answers three questions: *when can it be used?* ( `activation` ), *who can it affect?* ( `target` ), and *what happens?* ( `resolution` ). Giving the Scout several actions lets the simulator choose among them; it does not execute the entire list every turn.

### Read Thorn Strike

The Scout's attack — an action definition

```
{
  "kind": "action",
  "id": "thorn-strike",
  "name": "Thorn Strike",
  "activation": "action",
  "target": {
    "side": "enemy",
    "rangeFeet": 5
  },
  "resolution": {
    "kind": "attack",
    "attackBonus": 4,
    "damage": {
      "dice": {
        "count": 1,
        "sides": 6
      },
      "bonus": 2,
      "damageType": "piercing"
    }
  }
}
```

Here, `activation: "action"` spends an Action. The target is an enemy within 5 feet. An attack roll at +4 decides whether the target takes 1d6 + 2 piercing damage. The field `bonus` is a flat addition to damage, not another die.

### Turn an ability into a spell

Use `kind: "spell"` and write its `level`, `school`, and `components`. Set `concentration: true` only when casting it should begin concentration. The spell's name is descriptive: "Fireball" does not supply range, dice, or a saving throw. A limited spell also needs a resource payment, introduced in chapter 9.

| FIELD                   | TYPE             | REQUIRED | MEANING                                                                                                                                                                                          |
|-------------------------|------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>activation</code> | string or object | No       | <code>action</code> , <code>bonusAction</code> , <code>reaction</code> , <code>freeAction</code> , or <code>legendaryAction</code> . The object form adds <code>countsAs: "multiattack"</code> . |
| <code>delivery</code>   | weapon           | No       | Enables ordinary weapon behavior such as opportunity attacks and weapon masteries. Omit it for a general ability.                                                                                |
| <code>trigger</code>    | damage.before    | No       | Optional automatic trigger for a <code>reaction</code> with a non-area <code>self</code> target. Omitted triggers are manual reactions, not automatic defenses.                                  |
|                         |                  |          |                                                                                                                                                                                                  |

| FIELD              | TYPE                      | REQUIRED | MEANING                                                                                                                                                                                                                                                                                                                        |
|--------------------|---------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| target             | selector                  | No       | Top-level target selection. Omit it when a sequence/random node delegates target selection.                                                                                                                                                                                                                                    |
| resolution         | resolution node           | Yes      | Instructions for the action's attacks, saves, and effects. An empty or no-effect action may never be chosen.                                                                                                                                                                                                                   |
| resource           | object                    | No       | { "ref": "resource-id", "cost": 2 } for one payment pool.                                                                                                                                                                                                                                                                      |
| payments           | array, 1–16               | No       | Alternative payment modes. Use this instead of resource, never both.                                                                                                                                                                                                                                                           |
| level / spellLevel | integer 0–9               | No       | Spell level metadata. Either canonical spelling is accepted.                                                                                                                                                                                                                                                                   |
| school             | spell-school enum         | No       | abjuration, conjuration, divination, enchantment, evocation, illusion, necromancy, or transmutation. Allowed on spells and actions with spell: true. Inherited normally; otherwise omitted schools retain the historical evocation default. This supplies native spell metadata, not automatic school-specific class features. |
| concentration      | boolean                   | No       | Marks the spell/action as concentration; starting it replaces prior concentration.                                                                                                                                                                                                                                             |
| components         | array, 0–3 unique entries | No       | verbal, somatic, and/or material; spell definitions and actions with spell: true only. Empty means componentless. Omission inherits, otherwise preserving the historical verbal + somatic default.                                                                                                                             |
| magical            | boolean                   | No       | Classifies a non-spell supernatural ability for antimagic suppression. It does not make the action a spell or give it spell components. Spells are always magical, even if this field is false.                                                                                                                                |
| legendaryCost      | integer 1–3, default 1    | No       | Legendary points spent in addition to the chosen resource/payment cost. Ordinary Action, Bonus Action and Reaction are not consumed.                                                                                                                                                                                           |

**Reserved activation values:** `lairAction` and `outOfCombat` are named by the Beta v0.1 vocabulary but this engine build rejects them as `unsupported_runtime`. Do not publish packages that use them yet.

### Spell component presence

Verbal components participate in Silence and blocked-speech checks; Somatic components are subject to the Slow spell's failure roll. Any nonempty component list remains eligible for the normal Counterspell checks, including a Material-only spell. `components: []` removes those component-based interactions but does not bypass antimagic, concentration, action economy, or payment.

A verbal-only healing spell

```
{
  "kind": "spell", "id": "healing-word", "name": "Healing Word",
  "level": 1, "school": "abjuration", "activation": "bonusAction",
  "components": ["verbal"],
  "target": { "side": "ally", "rangeFeet": 60, "includeSelf": true, "includeDowned": true },
  "payments": [{ "id": "first", "mode": "spellSlot", "resource": "magic", "slotLevel": 1 }],
  "resolution": { "op": "heal", "amount": { "dice": { "count": 2, "sides": 4 }, "bonus": 3 } }
}
```

Declare the owner's `magic` slot pool separately. Component presence follows the 2024 casting rules; it does not infer hand occupancy, a focus, inventory, prices, or costly-material consumption. Material objects/cost declarations are not supported by this array. Referenced resolutions borrow the entry spell's components and payment; a payload reference does not cast the referenced spell a second time.

### Automatic defensive reactions

**Dawnward Ward's automatic reaction**

```
"activation": "reaction",
"trigger": "damage.before",
"target": { "side": "self" },
"resource": { "ref": "resolve", "cost": 2 },
"resolution": {
  "op": "temporaryHitPoints",
  "amount": { "dice": { "count": 1, "sides": 8 }, "bonus": 4 }
}
```

When the owner is about to receive a damage batch, it chooses the highest positive-value legal, affordable declared reaction. Grant order breaks equal-value ties. Normal incapacitation, reaction availability, spell/antimagic feasibility, and resource spending apply. Payment occurs before the payload, preventing a recursive damage callback from spending the same reaction again. Temporary HP can protect against that incoming batch and never stack with existing temporary HP. The trigger is before mitigation and can therefore also fire against damage that immunity later cancels.

This trigger currently supports only self-targeted defenses. It does not mean Shield's attack-roll trigger, Counterspell, protection of another creature, or arbitrary event reactions. Passive event rules are separate and do not spend a reaction unless an explicit executable action does so.

## 5. Choose targets and areas

A selector describes the legal recipients, not the damage. Keep a weapon's target as `enemy` ; for healing, choose `ally` and allow the owner or a downed ally explicitly when appropriate.

**A healing target — replace the action's target field**

```
"target": {
  "side": "ally", "rangeFeet": 30,
  "includeSelf": true, "includeDowned": true
}
```

This allows one ally within 30 feet, including the caster and a downed ally. It does not resurrect the dead. Set `maxTargets: 3` to select up to three such allies; use `mode: "all"` only when every legal recipient should be affected.

| FIELD                             | TYPE                           | REQUIRED | MEANING                                                                                                                                                                                                                 |
|-----------------------------------|--------------------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>side</code>                 | enum                           | No       | <code>self</code> , <code>ally</code> , <code>enemy</code> , or <code>any</code> .                                                                                                                                      |
| <code>mode</code>                 | enum                           | No       | <code>best</code> lets the simulator choose, <code>random</code> draws a legal target only during committed resolution, and <code>all</code> targets every legal candidate.                                             |
| <code>rangeFeet</code>            | integer 0–1,000                | No       | Maximum range in feet, in 5-foot increments.                                                                                                                                                                            |
| <code>maxTargets</code>           | integer 1–16, default 1        | No       | Top-level best-mode creature selectors only, without an area or self-only side. Select up to this many distinct recipients for one cast and one payment. Not valid on nested selectors or multi-target weapon delivery. |
| <code>area</code>                 | shape + sizeFeet [+ widthFeet] | No       | Top-level action selectors only. Requires mode: all and side enemy, ally, or any; omit rangeFeet. Uses the owner's root square as origin.                                                                               |
| <code>requiresSight</code>        | boolean                        | No       | Require normal Encountererra visibility.                                                                                                                                                                                |
| <code>requiresTargetSight</code>  | boolean, default false         | No       | The target must be able to see the owner. For a reciprocal visual gaze, set both this and requiresSight. Blindness, invisibility, lighting, and native special senses affect these independent checks.                  |
| <code>requiresOwnedGrapple</code> | boolean, default false         | No       | Require a capture owned by this actor. A requiresCondition rider must belong to that same hold; another captor's grapple or an unrelated incapacitating spell does not qualify.                                         |
| <code>includeDowned</code>        | boolean                        | No       | Allow downed combatants in the candidate set.                                                                                                                                                                           |
| <code>includeSelf</code>          | boolean                        | No       | Opt the owner into an ally or any candidate set. Defaults to false. All filters and area geometry still apply; use side self for an exclusively self-targeted action.                                                   |
| <code>reuseTargets</code>         | boolean                        | No       | For repeated/delegated selection, allow a combatant to be selected again.                                                                                                                                               |
| <code>maximumSize</code>          | size enum                      | No       | Reject candidates larger than this size.                                                                                                                                                                                |
| <code>requiresCondition</code>    | condition                      | No       | Only targets that currently have this condition.                                                                                                                                                                        |
| <code>excludesCondition</code>    | condition                      | No       | Reject targets that currently have this condition.                                                                                                                                                                      |
| <code>excludeCreatureTypes</code> | creature-type array            | No       | Reject one or more creature types.                                                                                                                                                                                      |

If there is no legal target, the action cannot be used, except where a random resolution explicitly provides a fallback.

## Bounded creature groups

`maxTargets` makes one cast resolve its payload once per selected recipient, in selected order. It does not cast a spell repeatedly, charge one slot per recipient, or create separate concentration slots. All recipients must be legal from the final casting square before payment. If a committed earlier payload changes a later recipient's eligibility, that recipient is skipped without replacement or another payment.

A group spell rolls independent healing dice for each recipient. Its random table/branch limit counts all recipients: two draws each for eight recipients use the sixteen-selection maximum. Nested sequence-step selectors are supported only for random-table payloads.

Selector defaults are enemy side, best mode, 5-foot range, sight not required, downed targets excluded, owner excluded from ally/any lists, and target reuse allowed. `includeDowned` includes downed heroes as possible targets, but does not override other targeting restrictions: dead, swallowed, or Ethereal-untargetable creatures remain excluded. Healing restores a downed hero's consciousness; temporary HP alone do not revive it. A heal operation cannot bring back a dead creature.

## Radius, cone, and line breath weapons

The Umbral Dragon breath showcase provides a playable starter with a default cone and an inherited line variant. In placement choices, use `"breath-shape": "line"` for the line. Both spend the same owned breath resource; a turn-start dice expression restores it on 5–6. That expression rolls even when the pool is already full.

The dragon's separate Thought-Shearing Breath remains a psychic cone whichever necrotic shape you choose. Both breaths spend the same recharge pool. See its complete package at the end for how the choice and resource are connected.

`shape` is `radius`, `cone`, or `line`. `sizeFeet` is the radius for a radius area and the forward length for a cone or line. Dimensions are integer multiples of 5, from 5 to 1,000 feet. A line optionally takes `widthFeet`, default 5; omit width for other shapes. The editor's Area shape dropdown preserves size and removes line-only width when switching shapes.

### A 30-foot cone breath selector

```
"target": {
  "side": "any",
  "mode": "all",
  "area": { "shape": "cone", "sizeFeet": 30 }
}
```

For a line, use `"area": { "shape": "line", "sizeFeet": 60, "widthFeet": 5 }`. For a radius, use `"area": { "shape": "radius", "sizeFeet": 30 }`. Connect this selector to a `savingThrow` resolution with damage and, if intended, `halfOnSave`.

Breath selectors exclude their owner's whole footprint. A creature is hit when at least one occupied square intersects the area's footprint, on the same spatial layer and with line of effect from the casting position. Sight is an additional opt-in restriction. `side: "any"` includes allies: friendly damage matters to the simulator's choice. Use enemy-only targeting only for an ability that intentionally spares allies.

Areas use Encounterra's two-dimensional grid, not three-dimensional volumes. A cone uses a 60-degree footprint; a line uses its forward length and width. These instantaneous selectors originate at the owner. For remote or persistent areas, use `createArea` instead. Nested or delegated shaped selectors remain unsupported.

## 6. Give creatures reusable abilities

A **grant** means “give this creature the mechanics from that definition.” A definition is a reusable description, not an ability that every creature automatically receives. Listing its reference in a creature’s `grants` array is what gives that creature the ability.

### Example: reusable protective bark

Add this ability to the starter’s definitions

```
{
  "kind": "ability",
  "id": "protective-bark",
  "name": "Protective Bark",
  "modifiers": [
    {
      "kind": "armorClass",
      "amount": 1
    }
  ]
}
```

Add this field to Bramble Scout — fragment

```
"grants": ["#protective-bark"]
```

The Scout’s `stats.armorClass` stays 13, and the granted modifier makes its effective AC 14. Another creature in the same package can grant `#protective-bark` too; each gets its own +1 AC. Without the grant, the ability remains available for reuse but does nothing for the Scout. Removing the grant removes that bonus.

### Actions, grants, and effects are different

| WRITE THIS  | USE IT FOR                       | WHAT HAPPENS                                                                                                                                          |
|-------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| actions     | Available attacks or spells      | List <code>#thorn-strike</code> to make that action available. The simulator can choose it when its targeting, action cost, and resources allow.      |
| grants      | Reusable mechanics               | Give the owner an ability or feat containing modifiers, rules, actions, resources, or further grants. A grant can also give a single action or spell. |
| applyEffect | A temporary effect during combat | An action or rule installs an effect on its recipient. Defining or permanently granting a temporary modifier attachment does not cast it.             |

You can write actions, rules, and modifiers directly on a creature when they are unique to it. Use a reusable ability when several creatures should share a trait, when a trait has several parts, or when a choice should give a whole set of mechanics. You do not need grants for every small definition.

### An ability can collect several mechanics — definition fragment

```
{
  "kind": "ability", "id": "forest-training", "name": "Forest Training",
  "actions": ["#thorn-strike"],
  "grants": ["#protective-bark"]
}
```

With that bundle defined, replace the Scout's direct `actions` and earlier `grants` lists with `grants: ["#forest-training"]`. It receives Thorn Strike and Protective Bark through the bundle. Do not keep both the direct action and the bundle just to “make sure”: grant each intended mechanic once.

A grant does not spend an Action, cast a spell, summon a creature, or copy another creature's statistics. Each receiving creature owns its own rules and resource pools. Temporary modifier effects use `applyEffect`; weapon mastery capabilities use `weapon.mastery`. Later chapters show both forms, and how `args` customizes a reusable grant.

## 7. Apply conditions and end them

After an attack hits or a save fails, use `applyCondition` to change what a target can do. Always choose an ending: a turn boundary, a repeat save, damage, concentration, or the encounter ending. "Restrained" alone does not say how long a binding lasts.

### Example: paralysis with an end-of-turn save

#### A failed save applies a condition with repeat saves

```
{
  "kind": "savingThrow",
  "ability": "constitution",
  "dc": 16,
  "onFailure": {
    "op": "applyCondition",
    "condition": "paralyzed",
    "duration": "repeatSaveEndOfTurn",
    "repeatSave": { "ability": "constitution", "dc": 16 }
  },
  "onSuccess": { "op": "noEffect" }
}
```

Place this resolution on an action. The first failed Constitution save applies Paralyzed; the target then repeats the save at the end of its turns. A success ends that condition. Add an independent turn cap only if the intended ability needs one.

Valid conditions are blinded, charmed, confused, deafened, frightened, grappled, incapacitated, invisible, paralyzed, petrified, poisoned, prone, restrained, slowed, dragon-slowed, speed-zero, stunned, unconscious, and cannot-regain-hit-points. `dragon-slowed` halves speed, prevents reactions, and permits an action or bonus action, not both. Unlike full `slowed`, it does not penalize AC/Dexterity saves, restrict attack count, or introduce spell failure.

### Condition durations

`instant`, `startOfSourceNextTurn`, `endOfSourceNextTurn`, `endOfTargetNextTurn`, `repeatSaveEndOfTurn`, `untilDamage`, `encounter`, and `concentration`. A repeating save needs `repeatSave: { "ability": "constitution", "dc": 16 }`.

### Turn caps, waking, and staged conditions

`maxTargetTurns` is an optional literal integer from 1 to 600 on `applyCondition` with `encounter`, `untilDamage`, or `repeatSaveEndOfTurn`. It expires at the Nth subsequent end-of-target-turn notification, before that turn's repeat save. It counts target turns, not source turns, rounds, or wall-clock minutes. Omit it for no independent cap; zero is invalid. Time continues in antimagic while magical repeat saves are suppressed. This adds no new concentration-duration combination.

For `unconscious` with `untilDamage`, `canBeShakenAwake: true` also permits an adjacent ally's wake action. Damage, waking, or the optional cap can end that source-owned sleep. Damage while the magical sleep is suppressed does not end its stored effect. Ordinary cures still remove it. Sleep does not automatically exclude Constructs or Undead: put those exclusions on its selector.

### A capped, wakeable sleep payload

```
{
  "op": "applyCondition", "condition": "unconscious",
  "duration": "untilDamage", "maxTargetTurns": 10,
  "canBeShakenAwake": true
}
```

`repeatSave.onFailureCondition` replaces the initial condition on the first failed repeat save. Success removes the initial stage. The final condition lasts until removed or encounter cleanup and has no further saves; it retains its magical source and can outlast the initiator's death. Removing the original stage prevents escalation, even if an unrelated source still restrains the target. Immunity to the final condition prevents the initial application. This one-step transition requires `repeatSaveEndOfTurn` and cannot combine with `maxTargetTurns`. It is not a general state-machine language or cross-encounter petrification persistence.

### Calcifying Cocoon — delayed petrification

```
{
  "op": "applyCondition", "condition": "restrained",
  "duration": "repeatSaveEndOfTurn",
  "repeatSave": {
    "ability": "constitution", "dc": 15,
    "onFailureCondition": "petrified"
  }
}
```

The Marbleweaver Moth package combines this cocoon with Binding Filament, Leaden Dust's narrow slowing, and a physical bite. It is an original quarry trapper with ordinary flight (no hover exemption). Bindings from different actions retain separate ownership.

For size-limited riders, `maximumSize` on `push`, `pull`, `grapple`, or `applyCondition` restricts only that operation. The same field on a selector instead excludes the entire target. Do not confuse these when authoring an attack that still damages larger creatures. A successful save with conditional riders should use an explicit `onSuccess` damage branch: `halfOnSave` alone divides damage but does not remove other operations from the failure payload.

## 8. Combine attacks, saves, and effects

A resolution is the action's instructions. An `attack` chooses hit or miss; a `savingThrow` chooses failure or success; `operations` performs several effects in order. A `sequence` combines complete resolutions, such as a bite followed by two claws.

### A save for half damage

#### Replace Thorn Strike's resolution to make a poison burst

```
"resolution": {
  "kind": "savingThrow", "ability": "constitution", "dc": 13,
  "onFailure": {
    "op": "damage", "damageType": "poison",
    "amount": { "dice": { "count": 2, "sides": 6 } }
  },
  "onSuccess": {
    "op": "damage", "damageType": "poison",
    "amount": { "kind": "divide", "operands": [
      { "kind": "dice", "dice": { "count": 2, "sides": 6 } }, 2
    ] }
  }
}
```

The failed save rolls 2d6 poison damage. The successful save rolls the damage and halves it, rounding down. This explicit pair of branches is useful when failure also applies a condition: put that condition only in `onFailure`. For a damage-only save, `damage` with `halfOnSave: true` is the shorter form.

### Two attacks using one Action

#### Repeat the Scout's attack payload — a new action definition

```
{
  "kind": "action", "id": "double-thorns", "name": "Double Thorns",
  "activation": { "economy": "action", "countsAs": "multiattack" },
  "target": { "side": "enemy", "rangeFeet": 5 },
  "resolution": {
    "kind": "sequence", "repeat": 2,
    "steps": [{ "kind": "reference", "ref": "#thorn-strike" }]
  }
}
```

Add `#double-thorns` to the Scout's actions to offer this option. Both attacks use the selected target. A reference here reuses the attack's *resolution*, not another Action or another resource payment. For weapon combinations that can move or change targets between attacks, see `attackSequences` in chapter 14.

Mixed damage and a wider critical range

Compound-damage attack

```
{
  "kind": "action",
  "id": "ember-cleaver",
  "name": "Ember Cleaver",
  "activation": "action",
  "target": {
    "side": "enemy",
    "mode": "best",
    "rangeFeet": 5,
    "requiresSight": true
  },
  "resolution": {
    "kind": "attack",
    "attackBonus": 7,
    "criticalThreshold": 19,
    "damage": [
      {
        "dice": { "count": 2, "sides": 6 },
        "bonus": 4,
        "damageType": "slashing"
      },
      { "value": 3, "damageType": "fire" }
    ]
  }
}
```

This attack deals both slashing and fire damage. Its `criticalThreshold: 19` makes a natural 19 or 20 critical. It rolls extra damage dice on a critical, but the +4 slashing bonus and flat 3 fire damage are not doubled.

| FIELD                                 | TYPE             | REQUIRED | MEANING                                                                                                                                                                                                |
|---------------------------------------|------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>operation shorthand</code>      | operation object | No       | <code>{ "op": "heal", "amount": 5 }</code> is normalized to an operations node.                                                                                                                        |
| <code>operations</code>               | node             | No       | Runs its ordered operations. Consecutive damage operations form a compound packet.                                                                                                                     |
| <code>attack</code>                   | node             | No       | <code>attackBonus</code> , optional <code>criticalThreshold</code> , <code>damage</code> , <code>onHit</code> , and <code>onMiss</code> . Critical hits double dice, not flat bonuses.                 |
| <code>attack.criticalThreshold</code> | integer 2–20     | No       | Optional minimum unmodified d20 for a critical hit. Default <code>20</code> ; use <code>19</code> for 19–20 or <code>18</code> for 18–20. A literal integer, not an expression or parameter reference. |
| <code>attack.attackType</code>        | melee   ranged   | No       | Defaults to melee; range does not determine attack type. Conditions and visibility affect both types. Ranged attacks also suffer close-range Disadvantage.                                             |
| <code>savingThrow</code>              | node             | No       | <code>ability</code> , <code>dc</code> , <code>damage</code> , <code>halfOnSave</code> , <code>onFailure</code> , and <code>onSuccess</code> . <code>save</code> is an accepted alias.                 |
|                                       |                  |          |                                                                                                                                                                                                        |

| FIELD                                          | TYPE                | REQUIRED | MEANING                                                                                                                                                                                                                                                                                  |
|------------------------------------------------|---------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>savingThrow.autoFailCreatureTypes</code> | creature-type array | No       | Optional types that automatically take the failure branch without consuming a save roll, inspiration, or save reactions. Damage still rolls normally. Replay records an automatic failure; <code>save.failure</code> fires, but <code>save.roll</code> and <code>d20.test</code> do not. |
| <code>sequence</code>                          | node                | No       | Runs ordered steps. The node or each wrapped step can have a bounded repeat expression; wrapped steps may delegate target timing and selector.                                                                                                                                           |
| <code>randomTable</code>                       | node                | No       | Selects one eligible weighted or dice-mapped outcome, with duplicate policy and fallback.                                                                                                                                                                                                |
| <code>randomBranch</code>                      | node                | No       | Runs then/else according to an integer probability.                                                                                                                                                                                                                                      |
| <code>reference</code>                         | node                | No       | <code>{ "kind": "reference", "ref": "#action-id" }</code><br>delegates to another executable definition.                                                                                                                                                                                 |

Sequences have at most 64 steps. Each repeat expression is clamped to 1–16 at execution. The compiler also requires proof that the entire action makes at most 16 random table/branch selections, including nested repeats and references; see the aggregate budget below. Recursive references are rejected. A resolution may have at most 12 parent-to-child edges from its root to any leaf, including sequence wrappers, attack/save branches, and expanded references.

A nested typed sequence applies its own repeat exactly once: an outer repeat of 2 containing an inner repeat of 3 executes the inner payload 6 times. A step envelope such as `{ "repeat": 3, "resolve": { "op": "noEffect" } }` supplies a separate repeat around its payload. Keep sequences shallow and avoid circular references.

A `reference` node executes the referenced resolution. It does not pay that action’s resource cost, apply its activation economy, or separately cast a spell. Put payment and concentration on the enclosing action.

**Critical-hit range:** set `criticalThreshold` inside each `kind: "attack"` resolution, not on the action or combatant. A die at or above the threshold automatically hits, even when its modified total is below AC. Natural 1 still misses; flat bonuses and reactive modifiers do not turn a lower natural die into a critical. The threshold belongs only to that attack node, including when it is nested or referenced; it does not modify other attacks, saving throws, or native abilities.

**2024 critical damage:** native and Homebrew attacks roll twice as many attack damage dice, independently; they do not multiply one rolled total. Flat damage and non-damage operations are not doubled. For example, `1d6 + 3` becomes `2d6 + 3`, while a flat `3` remains `3`. Extra attack dice, such as Sneak Attack or Divine Smite, also receive extra rolls. This follows the 2024 Critical Hits rules. Increasing the critical range only changes how often this damage rule applies.

Critical damage applies to damage operations in the selected on-hit branch. A nested attack makes its own critical decision; a nested saving throw deals its own damage without inheriting that critical. A hit against a Paralyzed or Unconscious target within 5 feet is also critical, but the condition does not turn a miss into a hit. Omit `criticalThreshold` for the normal natural-20 range.

## 9. Add limited uses and spell slots

A **resource** is a pool belonging to the creature: charges, daily uses, or spell slots. An action's `resource` or `payments` field says what using that action spends. Several actions can share one pool by referring to the same ID.

Add a resource to the Scout — creature fragment

```
"resources": [
  { "id": "focus", "name": "Focus", "maximum": 3, "refresh": "shortRest" }
]
```

Make Double Thorns spend one Focus — action fragment

```
"resource": { "ref": "focus", "cost": 1 }
```

The Scout can now use Double Thorns three times before it needs a short rest. Thorn Strike still costs no Focus. Use the bare resource ID `"focus"` here, not the definition reference `"#focus"`. If an ability bundle declares a resource, a grant gives that pool to its receiving creature.

| FIELD                | TYPE                  | REQUIRED  | MEANING                                                                                                                                                       |
|----------------------|-----------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>id</code>      | lowercase slug        | Yes       | Stable identifier used by payments and resource operations.                                                                                                   |
| <code>name</code>    | string                | No        | Display name.                                                                                                                                                 |
| <code>kind</code>    | uses   spellSlots     | No        | Defaults to uses for existing flat pools. spellSlots declares a real level-aware slot pool.                                                                   |
| <code>maximum</code> | integer 1–100,000     | Flat pool | Maximum flat uses; uses is a legacy alias. Forbidden on a spellSlots pool.                                                                                    |
| <code>slots</code>   | level-to-count object | Slot pool | A nonempty object with keys <code>"1"</code> through <code>"9"</code> and integer counts 0–100,000. Omitted levels have zero slots. Forbidden on a flat pool. |
| <code>refresh</code> | enum                  | No        | <code>shortRest</code> , <code>longRest</code> , <code>round</code> , or <code>never</code> . Default is longRest.                                            |

### One resource

JSON example

```
"resource": { "ref": "focus", "cost": 2 }
```

## Alternative payment modes

### JSON example

```
"payments": [
  { "id": "daily-free", "mode": "daily", "resource": "free-cast", "cost": 1 },
  { "id": "spell-slot", "mode": "spellSlot", "resource": "level-1-slots", "cost": 1 },
  { "id": "at-will", "mode": "atWill" }
]
```

The simulator chooses among the affordable payment alternatives. A cost of 2 requires two uses remaining; one is not enough. A non-at-will payment must identify a declared resource.

A `spellSlot` payment without `slotLevel` retains the legacy flat-pool behavior shown above. It does not spend a native spell slot. Shared resource IDs share uses; separate IDs provide independent per-spell uses.

## Leveled spell-slot pools

### A configured caster's shared slots

```
"resources": [{
  "id": "magic", "kind": "spellSlots",
  "slots": { "1": 4, "2": 3, "3": 2 }, "refresh": "longRest"
}]
```

### Explicit payment alternatives on a level-1 spell

```
"payments": [
  { "id": "first-level", "mode": "spellSlot", "resource": "magic", "slotLevel": 1 },
  { "id": "second-level", "mode": "spellSlot", "resource": "magic", "slotLevel": 2 }
]
```

Each alternative spends exactly one slot of the specified level. Omit `cost` or set it to 1. The spell must have a printed level of 1–9, and the payment cannot select a lower level. A slot pool requires `slotLevel`; a flat pool forbids it. Use explicit `payments`, not the single flat-resource shorthand. Exhausted levels disappear from the candidate list and remain rejected at resolution. Remaining slots persist between linked encounters.

For a spell that may use any of its owner's slots, write `{ "id": "slots", "mode": "anySpellSlot" }`. This exposes each nonzero declared slot rank at or above the spell's printed level, across every owned slot pool, including separately authored Pact-style pools. Omit `resource`, `resourceRef` and `slotLevel`; omit `cost` or use 1. It grants no slots. An owner without slots gets no casting option from this payment; combine it with a daily payment when appropriate. Upcasting still requires an explicit formula.

Action, Bonus Action, Reaction, free-action and Legendary Action spells share the native limit of one slot-funded spell per turn. This allowance resets at each creature's turn, not just the caster's turn. Independent daily or at-will alternatives do not spend a slot. Slot counts are authored snapshots, not class progression: use `shortRest` for a separately configured Pact-style pool. A higher-level payment does not infer extra damage, targets, or other effect scaling from a spell name.

## Legendary actions with an additional payment

A legendary action always spends `legendaryCost` from its owner's shared legendary-point pool. If it also declares `resource` or `payments`, that charge or slot is an additional cost. Both pools must be affordable; an exhausted secondary pool must not consume legendary points. An at-will payment removes only the secondary cost. It does not make the legendary action free or grant an ordinary-turn activation.

### Definition fragments: a two-point, second-level-slot legendary heal

```
{
  "kind": "monster", "id": "bell", "name": "Resonant Bell",
  "stats": { "hitPoints": 50, "legendaryActionUses": 3 },
  "resources": [{ "id": "magic", "kind": "spellSlots", "slots": { "2": 2 } }],
  "actions": ["#afterchime"]
},
{
  "kind": "spell", "id": "afterchime", "name": "Restoring Afterchime",
  "level": 1, "school": "abjuration", "activation": "legendaryAction",
  "legendaryCost": 2, "target": { "side": "self" },
  "payments": [{ "id": "second", "mode": "spellSlot", "resource": "magic", "slotLevel": 2 }],
  "resolution": { "op": "heal", "amount": { "dice": {
    "count": { "kind": "multiply", "operands": [2, { "kind": "castLevel" }] } },
    "sides": 8
  } } }
}
```

Put these fragments inside a package's `definitions` array. Each committed use above spends two legendary points and one second-level slot, then heals 4d8. A creature with three legendary points can use it once before its point pool refreshes. The two spell slots refresh on a long rest by default; the legendary-point refresh does not refill them.

### Recover two expended second-level slots

```
{ "op": "restoreResource", "resource": "magic", "slotLevel": 2, "amount": 2 }
```

`spendResource` and `restoreResource` use `slotLevel` for a leveled pool, with the amount interpreted as a count of slots. Spending is atomic at that level; restoration stops at the pool's declared maximum. These operations move resources, not cast spells, and do not themselves consume the per-turn spell allowance. Their action or rule must separately express any recovery budget and timing restriction. They are not automatic Arcane Recovery or Flexible Casting.

2024 spell-slot and casting rules.

## Spell Recall: sometimes keep the slot

Extend `srd:5.2.1/feat/boon-of-spell-recall` and grant the local feat to your caster. When the caster uses a level 1–4 spell slot, one d4 is rolled. A result equal to the *slot's* level means that slot is not expended. For example, Memory Spark below uses a second-level slot, so it keeps that slot on a 2, even though the spell's printed level is 1.

## Spell Recall workshop

```
[
  {
    "kind": "hero", "id": "echo-warden", "name": "Echo Warden",
    "stats": { "level": 19, "hitPoints": 100, "armorClass": 15,
      "abilityScores": { "intelligence": 20 } },
    "resources": [{ "id": "magic", "kind": "spellSlots", "slots": { "1": 3, "2": 2 } }],
    "grants": ["#recall-training"], "actions": ["#memory-spark", "#remembered-vigor"]
  },
  { "kind": "feat", "id": "recall-training", "extends": "srd:5.2.1/feat/boon-of-spell-recall" },
  {
    "kind": "spell", "id": "memory-spark", "name": "Memory Spark", "level": 1,
    "school": "evocation", "activation": "action",
    "target": { "side": "enemy", "rangeFeet": 60 },
    "payments": [{ "id": "second", "mode": "spellSlot", "resource": "magic", "slotLevel": 2 }],
    "resolution": { "kind": "attack", "attackType": "ranged", "attackBonus": 11,
      "onHit": { "op": "damage", "damageType": "fire", "amount": { "dice": { "count": 3, "sides": 8 } } } }
  },
  {
    "kind": "spell", "id": "remembered-vigor", "name": "Remembered Vigor", "level": 1,
    "school": "abjuration", "activation": "bonusAction",
    "target": { "side": "ally", "includeSelf": true, "includeDowned": true, "rangeFeet": 60 },
    "payments": [{ "id": "first", "mode": "spellSlot", "resource": "magic", "slotLevel": 1 }],
    "resolution": { "op": "heal", "amount": { "dice": { "count": 2, "sides": 4 }, "bonus": 5 } }
  }
]
```

A retained slot leaves the turn's spell-slot allowance available. The caster could then cast another slot-funded spell if it still has the required Action, Bonus Action or Reaction. Recall does not grant extra actions, create missing slots, or allow a caster to gamble on a second slot after already expending one that turn. Failed Recall expends the slot normally. Slots of level 5–9, cantrips, at-will casting and flat daily-use pools do not qualify; neither does spending a slot through a resource operation instead of casting a spell.

For an original creature trait, use `{ "kind": "spellSlotRecall" }` in `modifiers`. It is a permanent creature trait or grant, not a temporary effect. An optional `whenEquipment` filter can restrict it, but it has no `amount`, `dice`, weapon filter or adjustable chance. Multiple matching grants still roll only once per cast. This adds only the casting benefit: keep ability-score increases in your final stats. 2024 Boon of Spell Recall.

# Hit Dice and rest recovery

An explicit pool of eight d10 Hit Dice

```
"stats": { "hitPoints": 84, "hitDice": { "count": 8, "size": 10 } }
```

Hit Dice use the shared rest system, not a flat uses resource or damage expression. The simulator spends them one at a time during a short rest until HP is full or the pool is exhausted. Each die heals its roll plus the configured Constitution modifier, minimum 1 HP; placement Constitution overrides update that modifier without refilling spent dice. A successful long rest restores all spent Hit Dice. Rest-blocking curses still apply. Remaining dice persist through encounter-chain reconstruction.

Specify the pool explicitly when mirroring a class or monster: HP, size, CR, and a class-name label do not infer it. This field describes one die size, not mixed-size multiclass pools, automatic leveling, or a player-selected rest-spending budget. Use `count: 0` for a creature with no Hit Dice. 2024 Short Rest rules.

# 10. Calculate values with expressions

Start with literal numbers whenever they express your design. Use an expression for “1d8 + 3 healing,” “8 + proficiency + Charisma modifier,” or damage that grows with the chosen spell-slot level. An expression goes in a numeric field such as an operation’s `amount`, not in every field that happens to contain a number.

Healing 1d8 + 3 — an operation

```
{ "op": "heal", "amount": {  
  "kind": "dice", "dice": { "count": 1, "sides": 8 }, "bonus": 3  
} }
```

An ability `stat` expression reads its modifier: Strength 18 gives +4, not 18. The following tables and examples show the available inputs and arithmetic. Literal-only fields, such as `criticalThreshold` and modifier amounts, cannot use these expressions.

Numeric fields accept an integer or a typed expression. Use an integer for a fixed value and an expression when a value depends on dice, a stat, or a parameter.

| FIELD                                                       | TYPE    | REQUIRED | MEANING                                                                                                                                                                                                                                                                                           |
|-------------------------------------------------------------|---------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>integer</code>                                        | integer | No       | A constant, such as 7 or -2.                                                                                                                                                                                                                                                                      |
| <code>dice</code>                                           | object  | No       | <code>{ "kind": "dice", "dice": { "count": 2, "sides": 6 }, "bonus": 3 }</code> . A fixed count is 1–100; a bounded count formula may produce 0–100. Sides are a literal 2–100.                                                                                                                   |
| <code>castLevel</code>                                      | object  | No       | <code>{ "kind": "castLevel" }</code> reads the originating spell's explicit payment slot level, or its printed level for daily/at-will casting. Cantrips read zero. Allowed only in spell resolutions; references inherit the caller's casting context, not the referenced spell's printed level. |
| <code>stat</code>                                           | object  | No       | <code>subject</code> is <code>self</code> or <code>target</code> for currentHp, maxHp, and ability names. Level and proficiency currently read the owner. Ability names evaluate to modifiers, not scores.                                                                                        |
| <code>spellcastingModifier</code>                           | object  | No       | <code>{ "kind": "spellcastingModifier" }</code> reads the owner's modifier for this spell's <code>spellcastingAbility</code> : intelligence, wisdom or charisma. Use it in a spell definition, not a separate effect or event rule. It does not change the creature's other spells.               |
| <code>parameter</code>                                      | object  | No       | <code>{ "kind": "parameter", "param": "amount" }</code> reads a bound integer parameter.                                                                                                                                                                                                          |
| <code>add / subtract / multiply / divide / min / max</code> | object  | No       | Apply the named operation left-to-right to 1–64 operands. Division uses intermediate fractions and truncates the final result toward zero; dividing by zero currently evaluates to zero.                                                                                                          |

Keep expression nesting at 32 levels or fewer. Random amounts roll when the action is used. Use independent dice when you want separate rolls; multiplying one rolled result changes the distribution.

Numbers and the final result of an expression must fit a signed 32-bit integer (–2,147,483,648 to 2,147,483,647), as well as the field's own range. Keep formulas small enough to stay within these bounds.

## Upcast dice formulas

### Two independent d8s per casting level, plus a fixed 3

```
{
  "kind": "dice",
  "dice": {
    "count": { "kind": "multiply", "operands": [2, { "kind": "castLevel" }] },
    "sides": 8
  },
  "bonus": 3
}
```

A count formula uses constants, `castLevel`, and arithmetic. The compiler must prove that it remains within 0–100 for every casting level 0–9; use `max` with zero when subtracting a base level could otherwise become negative. Random, parameter-dependent, and combatant-stat counts are currently rejected. Fractions in a count truncate toward zero. A zero count rolls no dice; its flat bonus still applies.

The formula controls how many independent dice are rolled, not a multiplier applied to one roll. Critical damage doubles those dice once and leaves flat bonuses unchanged. Cast-level expressions also work in save DCs, attack bonuses, repeats, and operation amounts. Event rules and non-spell wrappers cannot borrow a spell's casting context. Dynamic counts are for general resolutions; `delivery: "weapon"` requires literal dice counts. This does not automatically infer a published spell's scaling rules, class progression, or spell school; author the latter explicitly with `school`.

### Save DC from an owner ability modifier

```
{
  "kind": "add",
  "operands": [
    8,
    { "kind": "stat", "subject": "self", "stat": "proficiency" },
    { "kind": "stat", "subject": "self", "stat": "charisma" }
  ]
}
```

## Magic Initiate: learn spells without changing your other magic

Create a local feat and choose the Cleric, Druid or Wizard list, plus Intelligence, Wisdom or Charisma. Select two different cantrips and one first-level spell from that list. Each selected local spell declares its membership with `spellLists`. This is authored membership, not a lookup that supplies a named spell's mechanics.

The feat grants both cantrips at will, and one free casting of the first-level spell per Long Rest. Short Rests do not restore that free casting. The first-level spell is always available without a preparation choice and may also use any owned spell slots. The free casting stays at level 1 and does not spend the turn's spell-slot allowance. All casts still obey their ordinary activation, component, concentration and targeting requirements.

Rune Student learns two original cantrips added to the Druid list for this Homebrew package, plus 2024 Cure Wounds. The cantrips and their list membership are Homebrew, not published Druid spells. Charisma 16 gives +3: Ember Needle attacks at +5, Dawn Chime has DC 13, and Cure Wounds heals 2d8 + 3 for the free casting or 4d8 + 3 using the student's second-level slot.

## Magic Initiate workshop

```
[
  {
    "kind": "hero", "id": "rune-student", "name": "Rune Student",
    "stats": { "level": 3, "hitPoints": 22, "armorClass": 12,
      "abilityScores": { "charisma": 16, "wisdom": 10 } },
    "resources": [{ "id": "magic", "kind": "spellSlots", "slots": { "1": 2, "2": 1 } }],
    "grants": ["#initiate"]
  },
  {
    "kind": "feat", "id": "initiate", "name": "Magic Initiate (Druid)",
    "extends": "srd:5.2.1/feat/magic-initiate",
    "magicInitiate": {
      "spellList": "druid", "spellcastingAbility": "charisma",
      "cantrips": ["#ember-needle", "#dawn-chime"], "level1Spell": "#cure-wounds"
    }
  },
  {
    "kind": "spell", "id": "ember-needle", "name": "Ember Needle",
    "level": 0, "school": "evocation", "spellLists": ["druid"],
    "spellcastingAbility": "wisdom", "components": ["verbal", "somatic"],
    "target": { "side": "enemy", "rangeFeet": 120 },
    "resolution": {
      "kind": "attack", "attackType": "ranged",
      "attackBonus": { "kind": "add", "operands": [
        { "kind": "stat", "stat": "proficiency" }, { "kind": "spellcastingModifier" }
      ] },
      "onHit": [{ "op": "damage", "amount": { "dice": { "count": 1, "sides": 10 } }, "damageType": "fire" }]
    }
  },
  {
    "kind": "spell", "id": "dawn-chime", "name": "Dawn Chime",
    "level": 0, "school": "evocation", "spellLists": ["druid"],
    "spellcastingAbility": "wisdom", "components": ["verbal", "somatic"],
    "target": { "side": "enemy", "rangeFeet": 60 },
    "resolution": {
      "kind": "savingThrow", "ability": "dexterity",
      "dc": { "kind": "add", "operands": [8,
        { "kind": "stat", "stat": "proficiency" }, { "kind": "spellcastingModifier" }
      ] },
      "onFailure": [{ "op": "damage", "amount": 4, "damageType": "radiant" }]
    }
  },
  {
    "kind": "spell", "id": "cure-wounds", "name": "Cure Wounds",
    "level": 1, "school": "abjuration", "spellLists": ["cleric", "druid"],
    "spellcastingAbility": "wisdom", "components": ["verbal", "somatic"],
    "target": { "side": "ally", "rangeFeet": 5, "includeSelf": true, "includeDowned": true },
    "resolution": { "op": "heal", "amount": { "kind": "add", "operands": [
      { "dice": { "count": { "kind": "multiply", "operands": [2, { "kind": "castLevel" }] }, "sides": 8 } },
      { "kind": "spellcastingModifier" }
    ] } }
  }
]
```

Put this array in your package's `definitions`. The feat changes the casting ability only for its granted copies. The original Wisdom-based templates remain unchanged. Use `spellcastingModifier` wherever that choice matters; a literal DC or a specific `stat: "wisdom"` stays fixed. A creature can learn the same spell separately with another ability.

Selected templates must be executable local `spell` definitions. Omit `resource`, `payments`, extra resources, grants, actions, choices, rules, modifiers and weapon delivery. Inline any `kind: "reference"` resolution so the selected casting ability applies throughout. Ordinary effect and summon operations retain their separately authored definitions. A native-only catalog name, descriptive note or out-of-combat spell is not an executable substitute.

You can take Magic Initiate again with a different list. Give each feat a distinct local ID; each gets an independent free casting. Mutually exclusive placement choices may offer same-list alternatives. For a later-level snapshot, replace one selection with a spell of the same level and list and save a new package version. Update cantrip dice and final stats as needed; this does not infer progression or feat prerequisites. 2024 Magic Initiate rules.

# 11. Make traits respond to events

Use a rule for a passive trait that reacts to something happening, instead of an action the creature chooses. Read a rule as: “On `event`, if `when` is true, perform `do`.” The `when` part is a **predicate**: a yes-or-no test.

The Scout regrows while alive and injured — creature fragment

```
"rules": [{
  "event": "turn.start",
  "when": { "all": [
    { "hitPoints": { "subject": "self", "operator": "above", "value": 0 } },
    { "hitPoints": { "subject": "self", "operator": "below", "value": 25 } }
  ] },
  "do": [{ "op": "heal", "target": "self", "amount": 4 }]
}]
```

Add that field to Bramble Scout. At the start of its turn, it regains up to 4 HP if alive and below 25 HP; healing still cannot exceed its maximum. The rule consumes no Action. Update the literal threshold if you change its HP, or use only the positive-HP check and let the healing cap handle full HP.

To reuse this trait, put its `rules` array in an `ability` definition and grant the ability. For “regenerate unless damaged by fire since last turn,” use the typed history predicate described below, then customize it with parameters in the next chapter.

A rule subscribes to one event, optionally filters it with `when`, applies a trigger limit, and executes ordered `do` operations. `operations` is accepted as an alias for `do`.

Rules listen for events belonging to the creature that owns them. Use `subject: "self"` or `"owner"`, not another creature or a global subscription. The event table below states which actions generate each notification.

Executable events are: `encounter.start`, `round.start`, `turn.start`, `turn.end`, `rest.short`, `rest.long`, `action.start`, `action.end`, `spell.cast`, `attack.roll`, `attack.hit`, `attack.miss`, `save.roll`, `save.success`, `save.failure`, `d20.test`, `damage.before`, `damage.after`, `damage.received`, `healing`, `movement`, `condition.applied`, `resource.changed`, `downed`, and `death`.

## Event coverage and recipients

In a rule, `owner` is the creature with the trait. `eventTarget` identifies the other participant, or the owner for a self-only event. Read the event’s row before using it: for example, a Homebrew save event belongs to the action’s owner, not automatically to the creature making the save.

| EVENTS                                                                     | WHEN IT HAPPENS  | RECIPIENT AND TIMING                                                                                                                       |
|----------------------------------------------------------------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <code>encounter.start</code>                                               | Encounter start  | The owner when an encounter starts; simply adding a definition to a package does not trigger it.                                           |
| <code>round.start</code> , <code>turn.start</code> , <code>turn.end</code> | The owner’s turn | The Homebrew owner. Round start is observed at its first turn in that round, not at a separate global initiative boundary.                 |
| <code>rest.short</code> , <code>rest.long</code>                           | Completed rest   | The owner after the matching resource refresh. A long rest emits <code>rest.long</code> , not an additional <code>rest.short</code> event. |
|                                                                            |                  |                                                                                                                                            |

| EVENTS                                                                                                    | WHEN IT HAPPENS                       | RECIPIENT AND TIMING                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>action.start</code> ,<br><code>action.end</code> , <code>spell.cast</code>                          | Homebrew actions only                 | The acting owner, with its primary target as <code>eventTarget</code> . Spell cast accompanies a declarative spell action. These are notifications, not cancellation/interception hooks.                                                                                                  |
| <code>attack.roll</code> , <code>attack.hit</code> ,<br><code>attack.miss</code>                          | Homebrew attack resolutions only      | The attacker after the D20 outcome is resolved, before the selected hit/miss branch. <code>eventTarget</code> is the defender; damage types describe the attack's hit payload, even on a miss.                                                                                            |
| <code>save.roll</code> , <code>save.success</code> ,<br><code>save.failure</code> , <code>d20.test</code> | Homebrew attack/save resolutions only | The action's owner, not automatically the creature rolling a save. <code>eventTarget</code> is the selected target. Native saves, concentration checks, and other native D20 calls do not universally dispatch these events.                                                              |
| <code>damage.before</code>                                                                                | Incoming damage                       | The active-form recipient, after source damage bonuses and before absorption, resistance, immunity, temporary HP, or HP application. <code>eventTarget</code> is the source when known, otherwise the recipient. Fires once per positive single/compound damage batch, not once per type. |
| <code>damage.received</code>                                                                              | Incoming damage                       | Recipient-only notification after mitigation and HP application, before final zero-HP processing. <code>eventTarget</code> is the source when known. A compound packet emits once with its incoming types.                                                                                |
| <code>damage.after</code>                                                                                 | Receiving or dealing damage           | Retains the legacy notification for the damage recipient and, for Homebrew damage operations, the distinct source. Use <code>damage.received</code> to test incoming damage only.                                                                                                         |
| <code>condition.applied</code>                                                                            | Accepted condition application        | The recipient, then the distinct source, after the condition has taken effect. <code>eventTarget</code> is the other participant, or the recipient when the source is unknown. A composite condition emits once, not once per condition bit.                                              |
| <code>healing</code>                                                                                      | HP recovery; temporary-HP caveat      | After an actual HP gain, the recipient, then the distinct healer, with the other participant as <code>eventTarget</code> . Self recovery emits once with self as <code>eventTarget</code> . This also covers built-in spells and abilities that use the normal HP recovery rules.         |
| <code>movement</code>                                                                                     | Movement or displacement              | The moved owner receives the map notification. Homebrew push/pull/teleport also notify a distinct initiating owner with the moved target as <code>eventTarget</code> .                                                                                                                    |
| <code>resource.changed</code>                                                                             | Homebrew resource operations only     | The owner after <code>spendResource/restoreResource</code> . Action payment, automatic recharge, and rest refresh are not general <code>resource.changed</code> subscriptions.                                                                                                            |
| <code>downed</code> , <code>death</code>                                                                  | Homebrew damage/kill operations only  | Damage operations notify the source about a target becoming non-alive; kill notifies the recipient and distinct source. These are not universal native death/downing listeners.                                                                                                           |
| <code>area.enter</code> , <code>area.leave</code>                                                         | Reserved global rules; rejected       | Use a zone's <code>onEnter</code> and <code>onExit</code> callbacks instead. Those callbacks belong to that particular authored area.                                                                                                                                                     |

`damage.received` is recipient-only; it is not an alias for `damage.after`. Dealing Fire damage must not count as receiving Fire damage or suppress the dealer's incoming-damage regeneration rule.

A `healing` event normally follows an actual HP gain. Full HP, prevented healing, or a dead recipient do not produce that recovery event. Self-healing notifies once. Homebrew temporary-HP operations also generate a legacy `healing` notification, even without an increase; do not use it as proof that current HP increased.

`damage.before` covers native weapons, spell attacks, saving-throw damage, and Homebrew damage as individual or compound damage packets. It does not fire for an empty batch or one containing only zero/non-positive amounts. Its typed context contains only positive incoming packets, even if immunity later prevents damage. A spell negated before damage

resolution does not reach this hook. Separate damage batches still produce separate events; self-damage does not duplicate delivery.

A `damage.before` rule can provide temporary HP before incoming damage arrives. It is a passive trait; unlike an action with `activation: "reaction"`, it does not spend a Reaction.

Damage events describe a damage resolution, not a guarantee of lost current HP. Immunity or temporary HP can prevent that loss. Use an HP predicate when current HP matters; these events do not let you rewrite or cancel a damage amount.

**A ward against the first incoming damage batch each turn**

```
{
  "event": "damage.before",
  "oncePer": "turn",
  "do": [
    { "op": "temporaryHitPoints", "amount": 3 }
  ]
}
```

Attach this rule to the owner's `rules` array. The ward uses normal temporary-HP replacement rules, not stacking, and it can trigger even against immune damage. Omit `oncePer` only if protection against every incoming batch is intended.

Condition application notifications

An accepted condition application notifies the recipient and distinct source; self-application notifies once. Immunity or protection that prevents application produces no event. Applying a condition the target already has can still notify. Removing a condition does not emit `condition.applied`.

The notification arrives after the condition takes effect. Removing an incapacitating condition afterward cannot undo concentration already lost. `removeCondition` removes matching conditions from every source, including a whole composite entry; removing Grappled also releases its hold. `hasCondition` tests current state, not the newly applied condition's identity. There is no condition-type event filter.

Conditions from a spell retain their magical origin, including with repeat saves. A creature knowing spells does not make all its other actions magical: mark a non-spell supernatural action with `magical: true`.

| FIELD                      | TYPE                | REQUIRED | MEANING                                                                                                                                                                           |
|----------------------------|---------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>true / false</code>  | boolean             | No       | An unconditional or disabled branch.                                                                                                                                              |
| <code>not</code>           | predicate           | No       | Logical negation.                                                                                                                                                                 |
| <code>all / and</code>     | 1–64 predicates     | No       | Every child must match.                                                                                                                                                           |
| <code>any / or</code>      | 1–64 predicates     | No       | At least one child must match.                                                                                                                                                    |
| <code>hasCondition</code>  | condition or object | No       | <code>subject</code> may be self/owner or target/eventTarget.                                                                                                                     |
| <code>hitPoints</code>     | object              | No       | Compare a subject's current HP with an integer using below, atMost, equal, atLeast, or above and their long-form aliases.                                                         |
| <code>eventOccurred</code> | object              | No       | Query an owner event since encounter.started or self.previousTurnStarted. The current typed where filter supports <code>damageType.in</code> or <code>damageType.inParam</code> . |

A damage-type filter must match the named event itself. A Fire attack hit cannot satisfy a Fire attack-miss predicate; incoming Fire cannot invent an attack event. History is recorded even without a rule subscribed to that event, and a compound source retains all its incoming damage types. Damage-type filters on events without typed damage context cannot match.

`oncePer` accepts turn, round, encounter, or rest scope. For more than one trigger, use `limit: { "uses": 2, "scope": "round" }`.

Avoid rules that trigger themselves, such as healing in response to `healing`. Circular event behavior is rejected, even with a trigger limit. Give repeatable traits deliberate limits and try them in combat with the other creatures you intend to use.

#### Only trigger while injured and not poisoned

```
"when": {
  "all": [
    { "hitPoints": { "subject": "self", "operator": "below", "value": 20 } },
    { "not": { "hasCondition": "poisoned" } }
  ]
}
```

## 12. Customize grants with parameters and choices

Parameters are named inputs to a reusable definition. They let you use the same trait with different numbers without copying its rules. `parameters` declares the inputs; a grant's `args` supplies their values.

## One regeneration trait, several strengths

### Parameterized regeneration rule

```
{
  "kind": "ability",
  "id": "regeneration",
  "name": "Regeneration",
  "parameters": {
    "amount": {
      "valueType": "integer",
      "minimum": 0
    },
    "blocked-by": {
      "valueType": "damageTypeList"
    }
  },
  "rules": [
    {
      "event": "turn.start",
      "when": {
        "all": [
          {
            "hitPoints": {
              "subject": "self",
              "operator": "above",
              "value": 0
            }
          },
          {
            "not": {
              "eventOccurred": {
                "event": "damage.received",
                "subject": "self",
                "since": "self.previousTurnStarted",
                "where": {
                  "damageType": {
                    "inParam": "blocked-by"
                  }
                }
              }
            }
          }
        ]
      },
      "do": [
        {
          "op": "heal",
          "target": "self",
          "amount": {
            "kind": "parameter",
            "param": "amount"
          }
        }
      ]
    }
  ]
}
```

This ability declares an integer healing `amount` and a list of damage types called `blocked-by`. Its rule skips regeneration if the owner received one of those types since its previous turn started. The amount expression reads the value bound by the grant.

### Parameterized grant

```
{
  "kind": "monster",
  "id": "bramble-scout",
  "name": "Bramble Scout",
  "stats": {
    "hitPoints": 25,
    "armorClass": 13,
    "speedFeet": 30,
    "size": "small",
    "creatureType": "plant"
  },
  "actions": [
    "#thorn-strike"
  ],
  "grants": [
    {
      "ref": "#regeneration",
      "args": {
        "amount": 4,
        "blocked-by": [
          "fire"
        ]
      }
    }
  ]
}
```

This Scout binds `amount` to 4 and `blocked-by` to `["fire"]`. Another creature can bind 8 and `["fire", "radiant"]`. Put the regeneration definition in the same package as the creature and its attack. Edit the existing Scout entry rather than appending a duplicate ID.

## Reusable parameters

Definitions may declare up to 64 parameters. `integer` parameters support minimum, maximum, and default. `damageTypeList` parameters support a default array. Grants must supply every parameter without a default and may forward a compatible parent parameter using `{ "param": "power" }`. Bound integer values are checked against each receiving definition's range after nested forwarding. Option grants declared on a parameterized bundle may forward that bundle's arguments; combatant parameters cannot be supplied directly at placement time.

## Forwarding an input through a bundle

### Pass a bundle's power into the child ability's amount — grant fragment

```
"grants": [{
  "ref": "#regeneration",
  "args": { "amount": { "param": "power" }, "blocked-by": ["fire"] }
}]
```

The granting bundle must declare an integer `power` parameter. If the creature grants that bundle with `"args": { "power": 6 }`, the regeneration ability receives `amount: 6`. Use explicit forwarding when names differ. A matching parent parameter of the same name and type can also supply an omitted argument; otherwise the child's default is used. A required input with neither source is an error.

## Placement choices

A choice declares a stable `id`, optional name, non-empty options, and a default option ID. Options may be strings or objects with `id`, optional `name`, and optional `grants`. Choose a different option in the placed creature's settings to override the default. Unknown IDs or option values are errors.

Option-specific grants make choices mechanical. Each selected option installs its referenced actions, spells, rules, and nested bundles, with ordinary typed grant arguments. Unselected options give none of those mechanics. A plain string or an object without grants is still metadata; an option ID such as `fire` does not itself change a damage type.

### A mechanical elemental choice — from Emberbound Adept

```
"choices": [{
  "id": "element", "name": "Bound element", "default": "fire",
  "options": [
    { "id": "fire", "grants": [{ "ref": "#fire-training", "args": { "accuracy": 7 } }] },
    { "id": "cold", "grants": [{ "ref": "#cold-training", "args": { "accuracy": 7 } }] }
  ]
}]
```

The complete Emberbound Adept package defines both training bundles. Choosing `cold` gives the cold spells; choosing `fire` gives the fire spells. Change the default in the definition, or choose a different option for the creature when placing it in the encounter. The choice does not change a damage type by itself.

## Combining choices and resource pools

- Grants accept local references such as `#ability-id` or objects with `ref` and `args`. An option cannot grant a monster/hero snapshot or bypass native-only restrictions.
- A granted bundle may contain another choice. Its mechanics require both the parent and child selections. All declared choices, including inactive branches, remain visible and retain defaults/pinned selections; changing an inactive child has no combat effect.
- Choice IDs and resource IDs must be unique across the combatant's entire expanded grant graph, including alternatives. Use distinct IDs for option-owned pools; repeated grants of a resource-owning or choice-owning

bundle still fail duplicate validation.

- Resources belonging to an unselected option are unavailable. A rest does not make an inactive option's pool usable.
- Every option is validated, not only the default. Bad references, mismatched arguments, out-of-range forwarded values, cycles, and excessive nesting reject the package before simulation.
- The graphical editor exposes each option's grants and integer, damage-type-list, or parent-parameter arguments. Reordering option IDs preserves existing grant data. Renaming an ID creates a new option; copy its grants explicitly if desired.

## 13. Reuse and adapt definitions

You have already used `#thorn-strike` to connect definitions. A reference reuses another part of the same package. Inheritance goes a step further: it starts a *new* definition from an existing one, then changes selected fields. Use inheritance for variants, and grants for giving mechanics to a creature.

### A stronger Thorn Strike — add a definition

```
{
  "kind": "action", "id": "heavy-thorn", "name": "Heavy Thorn",
  "extends": "#thorn-strike",
  "replace": { "resolution": { "attackBonus": 6, "damage": { "bonus": 4 } } }
}
```

Heavy Thorn keeps the parent's targeting and 1d6 piercing die, but attacks at +6 and adds 4 damage. Give the Scout `#heavy-thorn` in place of `#thorn-strike` to use the variant. Changing the child does not change its parent.

### Reference syntax

- `#ember-cleaver` refers to a definition in the same package.
- `srd:5.2.1/monster/commoner` refers to the locked public SRD catalog.
- Beta v0.1 intentionally has no references to other Homebrew packages. This keeps each package self-contained.

Local action references may point forward or backward in the definitions array. If an action omits its top-level target and delegates selection to a referenced random table, that delegation is resolved after the whole package is linked. Reordering definitions does not change the inferred targeting.

### Descriptive traits are not simulated mechanics

Grappler is documentation only in Beta v0.1. You may record it on a character, but the simulator does not apply its combined punch-and-grab, feat-specific Advantage, or grapple-carrying benefit, and does not actively plan tactical grappling for this feat. Existing monster grab/restrain actions are separate mechanics and are unchanged.

### Grappler documentation-only trait

```
{
  "kind": "feat", "id": "grappler-note",
  "name": "Grappler (documentation only)",
  "description": "Not simulated: Grappler feat benefits and tactical grappling are unavailable in Beta v0.1."
}
```

Add this local definition and grant `#grappler-note` to show the note in the stat block. Do not add `extends`, modifiers, actions or rules to this descriptive entry. The catalog's `srd:5.2.1/feat/grappler` listing is not an executable grant. Its score increase, if applicable, is already included in the final stats.

## Single-parent inheritance

`extends` accepts one local or SRD reference. The child may use the following controls before declaring ordinary child fields:

- `replace` recursively replaces object leaves and replaces scalar/array values.
- `add` appends arrays, merges objects, and otherwise assigns the provided value.
- `remove` removes matching array values or entries whose `id` / `ref` matches a string; `true` removes an object field.

Inheritance depth is limited to 32 and cycles are rejected. Most catalog entries currently use optimized native implementations and are marked `nativeOnly`; combatant entries can be aliased, but mechanical overrides are rejected. Alias names label the package entry; native combat narration may retain the preset name.

`srd:5.2.1/feat/defense`, `srd:5.2.1/feat/archery`, `srd:5.2.1/feat/great-weapon-fighting`, `srd:5.2.1/feat/savage-attacker`, and `srd:5.2.1/feat/two-weapon-fighting` are executable modifier definitions, not native-only aliases. Extend them and grant the resulting local feats. Their equipment requirements are mechanical, not inferred from names. Nick, Graze, Sap, Vex, Push, Slow, Topple and Cleave are executable weapon-bound masteries: extend the matching `srd:5.2.1/mastery/nick`, `srd:5.2.1/mastery/graze`, `srd:5.2.1/mastery/sap`, `srd:5.2.1/mastery/vex`, `srd:5.2.1/mastery/push`, `srd:5.2.1/mastery/slow`, `srd:5.2.1/mastery/topple` or `srd:5.2.1/mastery/cleave` locally and attach through `weapon.mastery`, not a global grant. Other catalog feat entries are not necessarily reusable Homebrew definitions. The compiler does not infer class progression or verify feat prerequisites for a level snapshot.

# 14. Equipment, weapons, and masteries

Use this chapter when a creature needs ordinary weapon behavior, such as opportunity attacks, multiple weapon attacks, Light, or a mastery. The SRD catalog exposes one reusable `weapon` definition for each supported weapon and magic bonus, such as `srd:5.2.1/weapon/greataxe-plus-1`. Extend that definition locally and reference it from any combatant's `actions`; its attack and damage expressions use that wielder's proficiency and ability scores. For an original weapon, write the fixed equipment facts and explicit formulas yourself—the compiler still does not infer mechanics from a display name.

## Ordinary weapons versus general attack resolutions

Our Scout's general attack resolution is enough for a simple damage ability. A `kind: "weapon"` definition always uses `delivery: "weapon"`; an ordinary `action` can opt into the same behavior explicitly. This enables opportunity attacks, long-range penalties, Light, and weapon-specific modifiers. The form has stricter requirements than a general resolution; use the table and restrictions below.

A limited-ammunition ranged weapon

```
{
  "kind": "action", "id": "javelin", "name": "Javelin",
  "activation": "action", "delivery": "weapon",
  "target": { "side": "enemy", "rangeFeet": 120, "includeDowned": true },
  "weapon": { "normalRangeFeet": 30, "ammunition": 1, "dangerZone": true },
  "resolution": {
    "kind": "attack", "attackType": "ranged", "attackBonus": 6,
    "onHit": { "op": "damage", "amount": {
      "dice": { "count": 2, "sides": 6 }, "bonus": 4
    }, "damageType": "piercing" }
  }
}
```

| FIELD                                 | TYPE              | REQUIRED | MEANING                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------|-------------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>weapon.normalRangeFeet</code>   | 5-foot increments | No       | Ranged normal range, no greater than <code>target.rangeFeet</code> . Beyond it, attacks have Disadvantage. Omitted: the target range. Melee reach uses native grid-hop distance and has no long-range penalty.                                                                                                                                              |
| <code>weapon.ammunition</code>        | integer 1–100,000 | No       | Omitted means unlimited. Each attack spends one ammunition use, including a miss. Omitted ammunition means unlimited uses.                                                                                                                                                                                                                                  |
| <code>weapon.opportunityAttack</code> | boolean           | No       | Expose this melee weapon as an opportunity attack, using the reaction budget. Omitted is false. Finite ammunition is shared with the ordinary attack, not duplicated for reactions; a miss spends ammunition too. Ranged opportunity attacks are rejected. An actor has one opportunity weapon; if several grants opt in, the last active grant selects it. |
| <code>weapon.dangerZone</code>        | boolean           | No       | Use this weapon for projected danger. Useful for finite ranged ammunition: an exhausted weapon no longer projects a threat it cannot deliver. The last active explicit danger-weapon grant wins; otherwise the opportunity weapon is used.                                                                                                                  |
|                                       |                   |          |                                                                                                                                                                                                                                                                                                                                                             |

| FIELD                               | TYPE                           | REQUIRED | MEANING                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------------------|--------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>weapon.attackAbility</code>   | strength / dexterity / finesse | No       | Explicit attack ability, separate from authored attack and damage expressions. Finesse chooses the wielder's higher Strength or Dexterity modifier when instantiated. Specify an ability for Light and masteries. This field does not grant a mastery.                                                                                                                               |
| <code>weapon.identity</code>        | lowercase slug                 | No       | Optional owner-local weapon identity. Omission uses this action definition's stable ID. Give melee/ranged variants or differently parameterized copies of one weapon the same explicit identity. Different definitions otherwise count as different weapons, regardless of display name. This does not create inventory or share ammunition between separately authored definitions. |
| <code>weapon.mastery</code>         | local or SRD reference         | No       | Optional reference to an executable local #mastery or srd:5.2.1/mastery/... definition. Nick, Graze, Sap, Vex, Push, Slow, Topple and Cleave require explicit attackAbility; Nick also requires Light and Cleave requires a melee attack. Global granting is rejected.                                                                                                               |
| <code>weapon.usesDex</code>         | boolean                        | No       | Legacy Dexterity-selection flag; omitted is false unless attackAbility selects Dexterity. If both fields are supplied, they must agree. Strength plus true, or Dexterity plus false, fails validation rather than choosing one silently.                                                                                                                                             |
| <code>weapon.twoHanded</code>       | boolean                        | No       | Whether the weapon is held in two hands. Defaults true when the twoHanded property is declared, false otherwise. A declared two-handed grip requires twoHanded or versatile in an explicit property list.                                                                                                                                                                            |
| <code>weapon.category</code>        | simple / martial / natural     | No       | Optional category fact. Omitted means unspecified, not an inferred category based on the weapon name.                                                                                                                                                                                                                                                                                |
| <code>weapon.type</code>            | melee / ranged                 | No       | Optional weapon classification, distinct from resolution.attackType. A thrown spear is a melee weapon making a ranged attack; a dart is a ranged weapon. Omission is unspecified and cannot match an explicit weapon-type requirement.                                                                                                                                               |
| <code>weapon.properties</code>      | unique string array, 0–9       | No       | ammunition, finesse, heavy, light, loading, reach, thrown, twoHanded, versatile. Omission preserves legacy unspecified properties; [] explicitly declares none. Light supplies the native Light classification, and twoHanded supplies the default grip.                                                                                                                             |
| <code>stats.attacksPerAction</code> | integer 1–16                   | No       | Default 1. Permits that many ordinary weapon attacks in one Attack action, replanning between each. Does not multiply generic resolution sequences. This is not an ordered bite-once/claw-twice composition language.                                                                                                                                                                |

The initial weapon form is deliberately constrained: an ordinary nonmagical action; enemy/best targeting with `includeDowned: true` and no extra filters or area; a single attack; no on-miss branch; stable additive attack bonus; additive damage dice/flat bonus; optional instantaneous, size-limited Prone riders. Secondary damage types currently accept dice without flat bonuses. Use ammunition instead of action resource/payments. Spell attacks, magical payloads, dynamic HP/target-dependent expressions, captures, saves, and on-zero/on-kill consequences must retain generic resolution. Unsupported combinations fail validation; no rider is silently discarded.

A resolution `reference` reuses only its referenced payload. It does not invoke that definition's weapon delivery, action economy, ammunition, trigger, or other top-level metadata. Grant the weapon action itself to use native weapon delivery; use generic referenced sequences for explicitly ordered compound abilities.

## Attack ability versus flat bonuses

Use `weapon.attackAbility` when the chosen Strength or Dexterity matters independently of the attack's final numbers. For Strength 18, the modifier is +4. A weapon with an additional +2 bonus can still have an explicitly authored flat damage bonus of +6; selecting Strength does not change it to +10. At Strength 7, the ability modifier is -2, including the rounding for odd scores. The editor selects the ability without changing your resolution payload.

Use `weapon.attackAbility` to identify the ability used by Light and supported masteries. All eight weapon masteries are available. Bind Cleave only to melee attacks. Two-Weapon Fighting is a separate feat grant; see the next workshop for restoring the Light extra attack's ability bonus.

Lowercase `strength`, `dexterity` and `finesse` are accepted inside a weapon-delivery definition's `weapon` object. Finesse selects the higher Strength or Dexterity score for that owner; ties use Strength. Other abilities, null, and a contradictory legacy `usesDex` fail validation. Inheritance, copies and opportunity attacks retain the choice. Owner-derived catalog formulas are evaluated when the combatant is instantiated; literal authored bonuses are never recalculated. The existing level-based proficiency calculation is unchanged and is not inferred from a monster's CR.

## Graze: damage on a missed weapon attack

Bind a local definition with `mastery: "graze"`, or extend `srd:5.2.1/mastery/graze`. An explicit `weapon.attackAbility` is required. Graze does not require Light and does not create an extra attack or once-per-turn resource. It applies to ordinary weapon attacks and their generated opportunity or Light-extra routes when their attack roll misses, including a natural 1. A rejected or out-of-range action is not a miss and deals no Graze damage.

The miss packet uses `max(0, floor((selected ability score - 10) / 2))` and the first weapon damage packet's type. At Strength 18 it deals 4, even if ordinary damage has a +6 flat bonus. Resistance halves it with rounding down, and immunity prevents it. Vulnerability, extra damage dice, weapon bonuses, on-hit riders and critical-hit scaling do not increase it. The 2024 Graze rule allows increases only through the attack's ability modifier. A miss stays a miss in the replay; actual damage still triggers the normal damage lifecycle, temporary-HP absorption, concentration checks and source attribution.

### Graze workshop

```
[
  {
    "kind": "hero", "id": "reaping-student", "name": "Reaping Student",
    "stats": { "level": 5, "hitPoints": 80, "armorClass": 16,
      "initiativeBonus": 20, "attacksPerAction": 2,
      "abilityScores": { "strength": 18 } },
    "actions": ["#training-scythe"]
  },
  { "kind": "mastery", "id": "graze", "extends": "srd:5.2.1/mastery/graze" },
  {
    "kind": "action", "id": "training-scythe", "name": "Training Scythe",
    "activation": "action", "delivery": "weapon",
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "weapon": { "attackAbility": "strength", "mastery": "#graze",
      "type": "melee", "category": "martial", "properties": ["twoHanded"] },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 9,
      "onHit": { "op": "damage", "damageType": "slashing",
        "amount": { "dice": { "count": 1, "sides": 8 }, "bonus": 6 } } }
  }
]
```

Put this list inside a package's `definitions`. On a miss, the student deals 4 slashing damage before any applicable reduction because its Strength modifier is +4. Change Strength to 16 to make the Graze amount 3; the explicitly written hit-damage bonus remains +6.

## Cleave: catch a second enemy with a melee weapon

Cleave turns a melee hit into an opportunity to attack a different enemy. The second enemy must be within 5 feet of the first and within the weapon's reach. This extra attack costs no additional Action, Bonus Action or Reaction, but it is available only once per turn across all of the wielder's weapons. It can trigger on another creature's turn through an Opportunity Attack. The first hit need not deal damage, and a killing blow still qualifies.

Attach a local `mastery: "cleave"` capability or extend `srd:5.2.1/mastery/cleave`, as below. Set `weapon.attackAbility` explicitly. The extra attack uses the same weapon's attack bonus, damage dice, additional damage packets and critical range. Following the 2024 Cleave rule, remove only a positive attack ability modifier from its damage; retain a negative modifier and any other flat bonuses. Cleave cannot trigger another Cleave on the same turn.

### Cleave workshop

```
[
  {
    "kind": "hero", "id": "ember-harvester", "name": "Ember Harvester",
    "stats": { "level": 5, "hitPoints": 80, "armorClass": 16,
      "attacksPerAction": 2, "abilityScores": { "strength": 18 } },
    "actions": [ "#ember-axe" ]
  },
  { "kind": "mastery", "id": "cleave", "extends": "srd:5.2.1/mastery/cleave" },
  {
    "kind": "action", "id": "ember-axe", "name": "Ember Axe",
    "activation": "action", "delivery": "weapon",
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "weapon": { "attackAbility": "strength", "mastery": "#cleave",
      "opportunityAttack": true, "type": "melee", "category": "martial",
      "properties": [ "heavy", "twoHanded" ] },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 9,
      "onHit": [
        { "op": "damage", "damageType": "slashing",
          "amount": { "dice": { "count": 1, "sides": 12 }, "bonus": 6 } },
        { "op": "damage", "damageType": "fire",
          "amount": { "dice": { "count": 1, "sides": 6 } } }
      ]
    }
  }
]
```

Put this list in a package's `definitions`. Place two enemies next to one another and within 5 feet of the harvester. A normal hit deals 1d12 + 6 slashing plus 1d6 fire. The Cleave hit deals 1d12 + 2 slashing plus 1d6 fire: Strength supplies +4 of the written +6, so only +4 is removed. A critical hit rolls twice the damage dice, without doubling flat bonuses. If you add `weapon.ammunition` as a finite attack-use counter, the extra attack spends one use too.

No legal second enemy means Cleave remains available for a later hit that turn. Cover, blocked reach and defensive reactions still apply to the extra attack. For a weapon that can also be thrown, make a separate ranged action sharing `weapon.identity` and leave Cleave off that ranged action.

## Topple: knock a target Prone

Bind a local `mastery: "topple"` definition or extend `srd:5.2.1/mastery/topple`. A hit forces a Constitution saving throw; failure applies Prone. Damage is not required, so a target immune to the weapon's damage can still be knocked down. Topple has no size limit, but immunity to Prone prevents the condition. It is a nonmagical weapon mastery, not a spell.

The save DC is **8 + the modifier of `weapon.attackAbility` + proficiency bonus**, following the 2024 Topple rule. Specify Strength or Dexterity explicitly. Neither `resolution.attackBonus`, the flat damage bonus nor `stats.difficultyClass` sets this DC. Proficiency follows `stats.level`: +2 at levels 1–4, +3 at 5–8, +4 at 9–12, +5 at 13–16 and +6 at 17–20. For a monster snapshot, set the level explicitly as well; its CR is a separate field.

### Topple workshop

```
[
  { "kind": "hero", "id": "stone-warden", "name": "Stone Warden",
    "stats": { "level": 5, "armorClass": 15, "hitPoints": 60,
      "speedFeet": 30, "size": "medium", "creatureType": "humanoid",
      "abilityScores": { "strength": 18, "dexterity": 12, "constitution": 16,
        "intelligence": 10, "wisdom": 12, "charisma": 10 } },
    "actions": [ "#training-staff" ] },
  { "kind": "mastery", "id": "topple", "extends": "srd:5.2.1/mastery/topple" },
  { "kind": "action", "id": "training-staff", "name": "Training Staff",
    "activation": "action", "delivery": "weapon",
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "weapon": { "attackAbility": "strength", "mastery": "#topple", "opportunityAttack": true },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 7,
      "onHit": { "op": "damage", "damageType": "bludgeoning",
        "amount": { "dice": { "count": 1, "sides": 6 }, "bonus": 6 } } } }
]
```

Put this list inside a package's `definitions`. The staff's Topple DC is 15: 8 + 4 Strength + 3 proficiency. Its hit damage is 1d6 + 6; that extra damage does not raise the save DC. Change Strength to 16 to lower the DC to 14, then update the explicitly written attack and damage bonuses if that is also intended.

The simulator attempts Topple automatically on a hit against a target that is not already Prone. A successful save allows another attempt on a later hit; there is no once-per-turn limit. Prone does not expire at the wielder's next turn: the target must stand or otherwise remove it. Melee and ranged weapon attacks can carry Topple, including generated opportunity and Light-extra attacks. Remove `weapon.mastery` to keep only the ordinary attack.

## Slow: non-stacking movement penalty

Bind a local `mastery: "slow"` capability or extend `srd:5.2.1/mastery/slow`. It requires explicit `weapon.attackAbility`, but not Light. A hit must deal actual damage, including damage absorbed by temporary HP or a surviving secondary damage type. A miss, immunity to the entire packet or complete damage reduction does not apply Slow. The target's walking, flying, swimming, climbing and burrowing speeds are reduced by 10 feet, each with a floor of zero, until the start of the wielder's next turn. This is a nonmagical mastery, not the *Slow* spell: it does not halve speed, reduce AC or remove reactions.

Multiple Slow weapons share one 10-foot reduction, not cumulative penalties. Each wielder retains an independent expiration clock; a fresh damaging hit refreshes only that wielder's marker. A second source or refresh does not deduct more movement. Temporary penalties do not rewrite base speeds, so later speed changes, typed walking-speed modifiers

and form changes remain intact on expiry. Slow applies to newly acquired movement modes too. The 2024 Speed rules explain special-speed changes.

### Slow workshop

```
[
  { "kind": "hero", "id": "trail-warden", "name": "Trail Warden",
    "stats": { "level": 5, "armorClass": 15, "hitPoints": 160,
      "speedFeet": 30, "initiativeBonus": 20, "size": "medium", "creatureType": "humanoid",
      "abilityScores": { "strength": 12, "dexterity": 16, "constitution": 16,
        "intelligence": 10, "wisdom": 12, "charisma": 10 } },
    "actions": [ "#training-bow" ] },
  { "kind": "mastery", "id": "slow", "extends": "srd:5.2.1/mastery/slow" },
  { "kind": "action", "id": "training-bow", "name": "Training Bow",
    "activation": "action", "delivery": "weapon",
    "target": { "side": "enemy", "mode": "best", "rangeFeet": 120, "includeDowned": true },
    "weapon": { "attackAbility": "dexterity", "mastery": "#slow", "normalRangeFeet": 120 },
    "resolution": { "kind": "attack", "attackType": "ranged", "attackBonus": 5,
      "onHit": { "op": "damage", "amount": { "dice": { "count": 1, "sides": 4 }, "bonus": 3 },
        "damageType": "piercing" } } }
]
```

Put this list inside a package's `definitions`. The training bow demonstrates a ranged Slow weapon. Remove `weapon.mastery` to keep the same ordinary attack without its slowing effect.

## Push: forced movement on a hit

Bind a local `mastery: "push"` capability or extend `srd:5.2.1/mastery/push`, with an explicit `weapon.attackAbility`. A weapon hit can push a Large-or-smaller target up to 10 feet straight away from the wielder. No damage is required: immunity, temporary HP or complete damage reduction do not cancel the hit rider. A miss, stale out-of-range action or unaffordable attack does not push. Huge and Gargantuan targets are not moved. This follows the 2024 mastery rules.

The simulator automatically uses the longest legal push, up to two grid cells, along the existing center-to-center grid direction. It stops at occupied cells, impassable terrain, map edges and applicable force barriers; it cannot jump over a blocker. Forced movement spends no target movement and does not provoke Opportunity Attacks. Generated opportunity and Light-extra routes retain the rider and their usual resource checks. A dead target is not pushed. Retaliation such as Fire Shield happens first: if it removes the wielder, no optional push follows, but the original hit and damage still count.

The encounter's chasm setting determines whether contact stops a ground target or kills it. Push is automatic up to the longest legal distance, not an author-configurable choice of destination. Flying targets use their movement and terrain restrictions.

## Push workshop

```
[
  { "kind": "hero", "id": "bridge-warden", "name": "Bridge Warden",
    "stats": { "level": 5, "armorClass": 15, "hitPoints": 160,
      "speedFeet": 30, "initiativeBonus": 20, "size": "medium", "creatureType": "humanoid",
      "abilityScores": { "strength": 16, "dexterity": 12, "constitution": 16,
        "intelligence": 10, "wisdom": 12, "charisma": 10 } },
    "actions": [ "#training-maul" ] },
  { "kind": "mastery", "id": "push", "extends": "srd:5.2.1/mastery/push" },
  { "kind": "action", "id": "training-maul", "name": "Training Maul",
    "activation": "action", "delivery": "weapon",
    "target": { "side": "enemy", "mode": "best", "rangeFeet": 5, "includeDowned": true },
    "weapon": { "attackAbility": "strength", "mastery": "#push",
      "opportunityAttack": true, "contributesToDangerZone": true },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 5,
      "onHit": { "op": "damage", "amount": { "dice": { "count": 1, "sides": 4 }, "bonus": 3 },
        "damageType": "bludgeoning" } } }
]
```

Put this list inside a package's `definitions`. This training maul deliberately has Push. Its name does not assign the mastery; `weapon.mastery` does. Remove the binding to keep only its normal attack.

## Sap and Vex: next-attack effects

Bind `mastery: "sap"` or `mastery: "vex"` through the same local-reference contract; neither requires Light. Both use the mastery effect on melee/ranged weapons and their generated opportunity and Light-extra routes. They are not global advantages, penalties, extra attacks, or once-per-turn resources. The 2024 Sap and Vex rules have different triggering and expiration conditions:

| MASTERY | TRIGGER                 | BEHAVIOR                                                                                                                                                                                                                                                                               |
|---------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sap     | hit; no damage required | The target has Disadvantage on its next attack roll before the start of the wielder's next turn. Immunity or a fully reduced damage packet does not prevent Sap on a hit. The target's own turn start does not expire it.                                                              |
| vex     | hit and actual damage   | The wielder has Advantage on its next attack roll against this target before the end of the wielder's next turn. Damage absorbed by temporary HP counts; immunity or reduction of the whole triggering packet to zero does not. A surviving extra damage type can satisfy the trigger. |

A miss applies neither effect. The next qualifying attack roll consumes its marker even if that roll misses, deals no damage, or Advantage and Disadvantage cancel. Saves do not consume either marker. Vex does not benefit another attacker, and attacking a different target does not consume it. A matching weapon, generic DEF attack, or spell attack can consume the effect; it need not use the original weapon. Repeated qualifying hits refresh that wielder's marker without stacking roll benefits; separate wielders retain separate expiration clocks. A rejected or out-of-range action makes no roll and neither consumes nor creates a marker.

## Sap and Vex workshop

```
[
  { "kind": "hero", "id": "dueling-student", "name": "Dueling Student",
    "stats": { "level": 5, "hitPoints": 160, "armorClass": 15,
      "speedFeet": 30, "initiativeBonus": 20, "attacksPerAction": 3,
      "abilityScores": { "strength": 16, "dexterity": 12,
        "constitution": 14, "intelligence": 10, "wisdom": 10, "charisma": 10 } },
    "actions": [ "#training-needle" ] },
  { "kind": "mastery", "id": "sap", "extends": "srd:5.2.1/mastery/sap" },
  { "kind": "mastery", "id": "vex", "extends": "srd:5.2.1/mastery/vex" },
  { "kind": "action", "id": "training-needle", "name": "Training Needle",
    "activation": "action", "delivery": "weapon",
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "weapon": { "attackAbility": "strength", "mastery": "#vex",
      "opportunityAttack": true, "dangerZone": true },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 5,
      "onHit": { "op": "damage", "damageType": "piercing",
        "amount": { "dice": { "count": 1, "sides": 4 }, "bonus": 3 } } } }
]
```

Put this list inside a package's `definitions`. The needle uses Vex. Change only its `weapon.mastery` to `"#sap"` to teach Sap instead. The three attacks and high HP are explicit practice values, not benefits inferred from the student's level.

## Light extra attacks, Nick and weapon identity

A weapon-delivery action with `properties: ["light"]` and explicit `attackAbility` creates a Light extra-attack route automatically. After an Attack-action attack with a different Light weapon on the owner's turn, the extra attack can spend the Bonus Action. A miss still qualifies. The extra attack removes only the positive ability contribution from base damage; negative ability modifiers and other flat bonuses remain. The example below has Dexterity 18 and a +2 weapon bonus: ordinary damage is 1d6 + 6, extra damage is 1d6 + 2. Extra damage dice and critical hits use the ordinary weapon damage rules.

Attach Nick to the weapon making the extra attack. Nick offers that same extra attack inside the Attack action without using another ordinary attack slot or the Bonus Action. Light and Nick share **one extra attack per turn** across every weapon and route; Extra Attack and Action Surge do not refresh it. The ordinary Light Bonus Action option remains available on a Nick weapon. Nick requires an ordinary Attack-action prerequisite, not Haste's single-attack action alone. Haste can enable the ordinary Light bonus attack. Slow's one-attack limit blocks Nick. Reactions, standalone spell attacks and another creature's turn do not establish a new prerequisite.

An extra attack cannot trigger another extra attack. Light requires an explicit attack ability so the format knows which positive ability contribution to remove from the authored damage bonus. Follow the Light and Nick equipment rules when designing a legal fixed loadout.

`weapon.identity` is a lightweight author assertion, not an inventory system. Matching explicit identities mean one weapon for Light's different-weapon check. Generated opportunity/Light/Nick routes preserve that identity and share their source action's finite ammunition. Separately authored actions do not acquire a shared ammunition pool merely by matching identities. Hands, drawing/stowing, transferring items, availability after throwing, and equipping proficiency are not simulated; provide a legal fixed snapshot.

Create a local mastery with `{"kind":"mastery","id":"nick","mastery":"nick"}`, or use the SRD inheritance shown here. Attach `"#nick"` through `weapon.mastery`. A mastery capability cannot carry parameters, resources, choices,

modifiers, rules, grants, actions, attachments or a resolution; use a separate granted ability for additional mechanics. All eight masteries use this weapon-bound contract. Two-Weapon Fighting is a separate feat grant, explained below.

### Light and Nick workshop

```
[
  {
    "kind": "hero", "id": "twin-blade-student", "name": "Twin-blade Student",
    "stats": { "level": 5, "hitPoints": 80, "armorClass": 16,
      "attacksPerAction": 2, "abilityScores": { "dexterity": 18 } },
    "actions": [ "#left-blade", "#right-blade", "#ward" ]
  },
  { "kind": "mastery", "id": "nick", "extends": "srd:5.2.1/mastery/nick" },
  {
    "kind": "action", "id": "left-blade", "name": "Moonsteel Blade",
    "activation": "action", "delivery": "weapon",
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "weapon": { "identity": "left-hand-blade", "attackAbility": "dexterity",
      "type": "melee", "properties": [ "light", "finesse", "opportunityAttack": true ],
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 9,
      "onHit": [ { "op": "damage", "damageType": "slashing",
        "amount": { "dice": { "count": 1, "sides": 6 }, "bonus": 6 } } ] }
    },
  },
  {
    "kind": "action", "id": "right-blade", "extends": "#left-blade",
    "replace": { "weapon": { "identity": "right-hand-blade", "mastery": "#nick" } }
  },
  {
    "kind": "action", "id": "ward", "name": "Moonward", "activation": "bonusAction",
    "target": { "side": "self" },
    "resolution": { "kind": "operations", "operations": [
      { "op": "temporaryHitPoints", "amount": 8 }
    ] }
  }
]
```

Put this list inside a package's definitions array. The two same-named blades are separate weapons. A legal first turn can make two ordinary attacks, one right-blade Nick attack after a left-blade attack, and Moonward with the preserved Bonus Action. Set the right blade's identity equal to the left blade's to model alternate attacks with just one weapon: neither can satisfy the different-weapon prerequisite. Remove its mastery binding to keep only the ordinary Light Bonus Action attack, competing with Moonward.

## Two-Weapon Fighting: restore the Light attack's ability bonus

Light normally leaves a positive ability modifier out of its extra attack's damage. The 2024 Two-Weapon Fighting feat restores it when it is not already included. This applies both to the Bonus Action route and to Nick. It does not give you another attack, change the different-weapon requirement, or add an ability modifier twice.

### Add this feat to the Light and Nick workshop's definitions

```
{
  "kind": "feat", "id": "two-weapon-training", "name": "Two-Weapon Fighting",
  "extends": "srd:5.2.1/feat/two-weapon-fighting"
}
```

### Grant the training to Twin-blade Student — creature fragment

```
"grants": ["#two-weapon-training"]
```

The preceding student has Dexterity 18 and +6 written damage. Without the feat, its Light or Nick extra attack deals 1d6 + 2; with this grant, it deals 1d6 + 6. Ordinary attacks already include the +4 ability modifier, so their damage stays unchanged. A negative modifier was already included and is not added again. Cleave remains a different extra attack and still omits a positive ability modifier, even when triggered by a Light attack.

For a custom ability or temporary effect, the equivalent modifier is `{ "kind": "lightExtraAttackAbility" }`. It is a flag, without `amount` or `dice`. Optional equipment and weapon filters narrow which Light extra attacks benefit. Multiple matching grants, including a native Two-Weapon Fighting feat, still restore the modifier only once.

## Lightweight equipment facts

| FIELD                                      | TYPE                          | REQUIRED | MEANING                                                                                                                                                                                    |
|--------------------------------------------|-------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>equipment</code>                     | hero/monster object           | No       | Only armor and shield are supported. There are no item IDs, inventory slots, prices, weights, proficiency rules, equipping actions or hand-occupancy simulation.                           |
| <code>equipment.armor</code>               | none / light / medium / heavy | No       | Worn armor category. none is explicitly unarmored; natural armor is part of base AC, not worn armor. Omission is unknown and never satisfies an armor filter.                              |
| <code>equipment.shield</code>              | boolean                       | No       | True means using a shield; false explicitly means no shield. Omission is unknown, not false. A shield does not count as worn armor for Defense.                                            |
| <code>modifier.whenEquipment.armor</code>  | 1–4 unique categories         | No       | Matches any listed worn-armor category on the modifier recipient. Supported on every permanent or temporary typed modifier; omitted means no armor requirement.                            |
| <code>modifier.whenEquipment.shield</code> | boolean                       | No       | Requires the recipient's explicit matching shield fact. With an armor filter, both requirements must match. Empty filter objects, unknown fields and duplicate categories fail validation. |

These fields describe a fixed combat snapshot. They do not infer AC, attack/damage modifiers, Dexterity selection from Finesse, Heavy ability-score penalties, Loading limits, Versatile damage dice, reach/ranges, ammunition recovery, or other mastery riders. Light extra attacks require the explicit ability contract above. Author the currently supported mechanics explicitly; missing property-specific automation remains a Beta limitation, not an implied rule implementation. The existing `usesDex` and `twoHanded` fields retain their saved meanings. Copies and opportunity routes retain the weapon facts.

Dawnward Sentinel demonstrates heavy armor and shield use: base AC 18 plus its armor-gated modifier is AC 19. Changing only the equipment facts does not remove the shield's already-authored +2 from base AC; update both when designing another snapshot. Directly overriding `equipment` or `weapon` replaces that whole object; use nested `replace` for a partial inherited change.

Armor-gated Defense snapshot

```
[
  {
    "kind": "hero", "id": "armored-guardian", "name": "Armored Guardian",
    "stats": { "level": 8, "hitPoints": 60, "armorClass": 18 },
    "equipment": { "armor": "heavy", "shield": true },
    "grants": ["#defense"], "actions": ["#blade"]
  },
  {
    "kind": "feat", "id": "defense",
    "extends": "srd:5.2.1/feat/defense"
  },
  {
    "kind": "action", "id": "blade", "name": "Guardian Blade",
    "delivery": "weapon",
    "weapon": { "category": "martial", "properties": ["versatile"], "twoHanded": false },
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "resolution": { "kind": "attack", "attackBonus": 5,
      "damage": { "damageType": "slashing", "amount": { "dice": { "count": 1, "sides": 6 }, "bonus": 3 } }
    }
  }
]
```

Put these definitions in a normal package envelope. The guardian's final AC is 19. Defense's armor requirement follows the 2024 rule; this example assumes the snapshot is eligible for the feat.

Weapon-specific modifiers and portable fighting styles

`weaponAttackBonus` adds a signed flat attack-roll bonus; `weaponDamageDieMinimum` floors each die in the first weapon damage-dice entry, including the extra copies on a critical hit. Neither applies to generic attack nodes, unarmed attacks, or spell attack rolls. They also apply to that weapon's opportunity attacks. Multiple attack bonuses add; multiple die floors take their maximum, not their sum. Flat damage and secondary damage entries are not floored. A floor does not reroll dice or consume extra randomness.

| FIELD                                 | TYPE                  | REQUIRED | MEANING                                                                                                                   |
|---------------------------------------|-----------------------|----------|---------------------------------------------------------------------------------------------------------------------------|
| <code>whenWeapon.type</code>          | melee / ranged        | No       | Require the weapon's declared classification, not its name or the attack's delivery mode.                                 |
| <code>whenWeapon.twoHanded</code>     | boolean               | No       | Require the actual grip. A versatile weapon used one-handed fails a true requirement.                                     |
| <code>whenWeapon.propertiesAny</code> | 1–9 unique properties | No       | At least one listed property must be explicitly present. Omitted/empty weapon properties cannot satisfy this requirement. |

These two kinds, `weaponDamageRollTwice`, and `lightExtraAttackAbility` accept `whenWeapon`. All supplied requirements must match, along with any `whenEquipment` armor/shield guard on the recipient. An empty filter is invalid;

omit it for an unrestricted weapon modifier. The filters do not grant items, weapon proficiency or a hand-management system. Native presets keep their existing style rules; a portable style requiring explicit facts will not infer a native weapon's missing facts from its name.

Archery adds +2 with Ranged weapons, not with a thrown Melee weapon. Great Weapon Fighting needs a Melee weapon, two-handed grip, and the Two-Handed or Versatile property. Its minimum face is 3. The official clarification excludes extra spell damage such as Divine Smite. In this constrained weapon format, the first dice entry is the weapon dice; keep add-on damage separate rather than merging it into that entry.

## Portable weapon-style workshop

```
[
  {
    "kind": "hero", "id": "weapon-student", "name": "Weapon Student",
    "stats": { "level": 8, "hitPoints": 80, "armorClass": 16,
      "abilityScores": { "strength": 16, "dexterity": 16 } },
    "equipment": { "armor": "heavy", "shield": false },
    "grants": ["#archery", "#great-weapon-fighting", "#savage-attacker"],
    "actions": ["#bow", "#maul", "#spear"]
  },
  {
    "kind": "feat", "id": "archery",
    "modifiers": [{ "kind": "weaponAttackBonus", "amount": 2,
      "whenWeapon": { "type": "ranged" } }]
  },
  {
    "kind": "feat", "id": "great-weapon-fighting",
    "modifiers": [{ "kind": "weaponDamageDieMinimum", "amount": 3,
      "whenWeapon": { "type": "melee", "twoHanded": true,
        "propertiesAny": ["twoHanded", "versatile"] } }]
  },
  {
    "kind": "feat", "id": "savage-attacker",
    "modifiers": [{ "kind": "weaponDamageRollTwice" }]
  },
  {
    "kind": "action", "id": "bow", "name": "Practice Bow", "delivery": "weapon",
    "weapon": { "type": "ranged", "category": "martial",
      "properties": ["ammunition", "twoHanded"], "attackAbility": "dexterity" },
    "target": { "side": "enemy", "rangeFeet": 60, "includeDowned": true },
    "resolution": { "kind": "attack", "attackType": "ranged", "attackBonus": 5,
      "damage": { "damageType": "piercing",
        "amount": { "dice": { "count": 2, "sides": 6 }, "bonus": 3 } } }
  },
  {
    "kind": "action", "id": "maul", "name": "Practice Maul", "delivery": "weapon",
    "weapon": { "type": "melee", "category": "martial",
      "properties": ["twoHanded"], "attackAbility": "strength" },
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 5,
      "damage": { "damageType": "bludgeoning",
        "amount": { "dice": { "count": 2, "sides": 6 }, "bonus": 3 } } }
  },
  {
    "kind": "action", "id": "spear", "name": "Thrown Practice Spear", "delivery": "weapon",
    "weapon": { "type": "melee", "category": "simple",
      "properties": ["thrown", "versatile"], "twoHanded": false,
      "attackAbility": "strength" },
    "target": { "side": "enemy", "rangeFeet": 60, "includeDowned": true },
    "resolution": { "kind": "attack", "attackType": "ranged", "attackBonus": 5,
      "damage": { "damageType": "piercing",
        "amount": { "dice": { "count": 1, "sides": 6 }, "bonus": 3 } } }
  }
]
```

Wrap the workshop in a package envelope. These are illustrative training weapons, not inferred SRD equipment statistics. The student explicitly has two styles and Savage Attacker; no class progression or feat eligibility is inferred. The bow attacks

at +7; the maul and thrown spear at +5. Maul dice cannot be below 3; bow and spear dice have no floor. All three qualify for the shared once-per-turn dice selection. No shield is declared with the two-handed weapons.

### Once-per-turn weapon dice selection (Savage Attacker)

`{ "kind": "weaponDamageRollTwice" }` implements the dice-selection mechanic of 2024 Savage Attacker. On the first eligible weapon hit of each turn, the simulator rolls the first weapon damage-dice packet twice and chooses the higher total. Melee, ranged and thrown weapon attacks qualify; generic resolution nodes, spell attack rolls and unarmed attacks do not. This is a flag: do not supply `amount`, `dice`, a custom use count or a reset field.

A critical includes the extra weapon dice in *each* candidate packet. For a 2d6 weapon, compare two 4d6 totals on a critical, not two independently chosen 2d6 totals. The flat damage bonus is added once. Secondary damage entries, class riders and their critical copies are rolled normally, outside the selection. A matching `weaponDamageDieMinimum` floors every die in both candidate weapon packets.

One use belongs to the recipient for the current global turn, shared by every matching attachment, weapon and native Savage Attacker grant. Misses and ineligible weapons do not spend it. Reapplying an effect, switching weapons, or adding another grant cannot refresh it. It resets when any combatant ends its turn, so an opportunity attack on another creature's turn may benefit again. Temporary magical grants follow suppression and expiry normally; suppression does not refund a spent use.

The simulator uses `weaponDamageRollTwice` automatically on the first eligible hit, rather than holding it for a later hit. Keep only the weapon's own dice in the first damage entry; put additional damage in separate entries.

### Alert: initiative training

A feat can change how the encounter begins without adding an action. This level-5 hero has a base Initiative bonus of +2; granting Alert adds its +3 proficiency bonus, for +5 in total. Keep `stats.initiativeBonus` as the base bonus; do not include Alert there again. Proficiency is derived from `stats.level`, not the descriptive challenge-rating label.

Alert workshop

```
[
  {
    "kind": "hero", "id": "dawn-watch", "name": "Dawn Watch",
    "stats": { "level": 5, "hitPoints": 38, "armorClass": 14,
      "initiativeBonus": 2, "abilityScores": { "dexterity": 14 } },
    "grants": ["#alert-training"], "actions": ["#watch-strike"]
  },
  { "kind": "feat", "id": "alert-training", "extends": "srd:5.2.1/feat/alert" },
  {
    "kind": "action", "id": "watch-strike", "name": "Watch Strike",
    "activation": "action", "delivery": "weapon",
    "weapon": { "attackAbility": "dexterity" },
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 5,
      "onHit": [{ "op": "damage", "damageType": "piercing",
        "amount": { "dice": { "count": 1, "sides": 6 }, "bonus": 2 } } ] }
  }
]
```

Alert also permits an Initiative exchange with a willing ally in the same encounter, as long as neither is Incapacitated. Encounterra treats teammates as willing and automatically exchanges with a slower ally when that ally has greater attack threat. Each combatant participates in at most one automatic exchange during the initiative pass. The final order may therefore differ from the individual rolled totals. This is not a Bonus Action or Reaction.

For your own traits, the equivalent flags are `{ "kind": "initiativeProficiency" }` and `{ "kind": "initiativeSwap" }`. You can grant either separately. Multiple matching proficiency grants still add the bonus once. They accept optional `whenEquipment` filters, but no `amount`, `dice` or weapon filter. These two modifiers must be permanent creature traits or grants: a temporary effect applied during combat cannot retroactively alter the opening initiative roll.

## Combat Prowess: rescue a missed attack

Extend `srd:5.2.1/feat/boon-of-combat-prowess` and grant the local feat. A missed weapon or spell attack can become a hit. The benefit then remains spent until the start of the creature's own next turn; an enemy's turn does not refresh it. A natural 1 can be rescued, but its natural die does not become a 20. This is separate from an increased critical-hit range.

### Combat Prowess workshop

```
[
  {
    "kind": "hero", "id": "steadfast-duelist", "name": "Steadfast Duelist",
    "stats": { "level": 19, "hitPoints": 120, "armorClass": 17,
      "attacksPerAction": 2, "abilityScores": { "dexterity": 20 } },
    "grants": ["#peerless-aim"], "actions": ["#rapier"]
  },
  { "kind": "feat", "id": "peerless-aim", "extends": "srd:5.2.1/feat/boon-of-combat-prowess" },
  {
    "kind": "action", "id": "rapier", "name": "Rapier",
    "activation": "action", "delivery": "weapon",
    "weapon": { "attackAbility": "dexterity", "opportunityAttack": true },
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 11,
      "onHit": [{ "op": "damage", "damageType": "piercing",
        "amount": { "dice": { "count": 1, "sides": 8 }, "bonus": 5 } } ] }
  }
]
```

The Duelist can rescue one miss across its attacks and opportunity attacks until its next turn starts. The grant adds neither an attack nor a Reaction. It also works on a spell's attack roll, but cannot rescue a failed saving throw or ability check. The simulator activates it on a missed roll after the attack's bonus and penalty dice. Multiple matching grants share the same use.

For an original trait, the equivalent modifier is `{ "kind": "combatProwess" }`. It accepts an optional `whenEquipment` filter, but no amount, dice, weapon filter or adjustable use count. It must be permanent, not a temporary effect. As with other feats, put any ability-score increase in the final stats. 2024 Boon of Combat Prowess.

## Fate: influence a nearby roll

Unlike Combat Prowess, Fate uses dice rather than guaranteeing a hit. Grant the local feat below to let the Threadkeeper influence its own rolls or those of other creatures within 60 feet. The grant supplies its own limited use: do not add a separate resource pool or an automatic event rule.

## Fate workshop

```
[
  {
    "kind": "hero", "id": "threadkeeper", "name": "Threadkeeper",
    "stats": { "level": 19, "hitPoints": 120, "armorClass": 17,
      "abilityScores": { "dexterity": 18, "wisdom": 20 } },
    "grants": ["#fate-training"], "actions": ["#threadblade"]
  },
  { "kind": "feat", "id": "fate-training", "extends": "srd:5.2.1/feat/boon-of-fate" },
  {
    "kind": "action", "id": "threadblade", "name": "Threadblade",
    "activation": "action", "delivery": "weapon",
    "weapon": { "attackAbility": "dexterity", "opportunityAttack": true },
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 10,
      "onHit": [{ "op": "damage", "damageType": "piercing",
        "amount": { "dice": { "count": 1, "sides": 8 }, "bonus": 4 } } ] }
  }
]
```

The simulator adds 2d4 to a failed allied test, or subtracts 2d4 from a successful enemy test, when that adjustment could change the outcome. It decides after the ordinary bonus and penalty dice, but before seeing Fate's dice. An insufficient result still spends the use. When several creatures could intervene, higher Initiative goes first, with a stable tie-break; one intervenes on that test.

Fate needs neither sight nor a Reaction. Its shared use refreshes after the owner rolls Initiative or completes a Short or Long Rest, not each turn. Initiative itself has no success or failure for Fate to change. Natural automatic outcomes stay automatic: Fate cannot turn an attack's natural 1 into a hit or cancel a natural critical. Ordinary saves and ability checks use their totals.

For an original trait, use `{ "kind": "boonOfFate" }`. It is a permanent modifier, optionally restricted by `whenEquipment`. Duplicate grants share the same use; removing and restoring an equipment requirement does not refill it. Amounts, custom dice, ranges, weapon filters and temporary effects are not supported for this preset. Keep ability-score increases in the final stats. 2024 Boon of Fate.

## Truesight: seeing an otherwise unseen target

The shared Truesight feat adds a 60-foot sense, not an attack bonus. Clear-Eyed Watcher can see an Invisible target within that range and attack it without the penalty for an unseen target. Walls and unrelated Blindness still block sight. No ability score is increased by the grant.

## Truesight workshop

```
[
  {
    "kind": "hero", "id": "watcher", "name": "Clear-Eyed Watcher",
    "stats": { "hitPoints": 95, "armorClass": 16, "speedFeet": 30,
      "level": 19, "abilityScores": { "dexterity": 20, "wisdom": 16 } },
    "grants": ["#clear-sight"], "actions": ["#watcher-bolt"]
  },
  {
    "kind": "feat", "id": "clear-sight", "name": "Clear Sight",
    "extends": "srd:5.2.1/feat/boon-of-truesight"
  },
  {
    "kind": "action", "id": "watcher-bolt", "name": "Watcher Bolt",
    "target": { "side": "enemy", "rangeFeet": 120, "requiresSight": true },
    "resolution": { "kind": "attack", "attackType": "ranged", "attackBonus": 11,
      "damage": { "dice": { "count": 1, "sides": 8 }, "bonus": 5, "damageType": "piercing" } }
  }
]
```

The bolt has 120-foot range, but Truesight reveals Invisible targets only within 60 feet. It also works in ordinary darkness, bypasses Mirror Image, and prevents the blindness imposed by the modeled Darkness spell. Duplicate grants do not extend the range or reduce a longer native sense. For an original trait, use `{ "kind": "boonOfTruesight" }`; optional `whenEquipment` guards work as usual. Custom ranges, temporary attachments and amount/dice fields are not supported for this preset.

Beta's Darkness spell uses an area-based blindness approximation, not opaque-volume ray tracing. Truesight does not bypass fog or unrelated Blinded conditions. Seeing into the Ethereal Plane, recognizing a transformed creature's true form, and arbitrary visual-illusion saving throws are not modeled; adding this feat does not add those systems. 2024 Boon of Truesight.

## Dimensional Travel: move after an action

A grant can also add a benefit that happens after another action. Rift Runner's Blink Steps needs no separate action definition or resource. Grant the feat, then give the creature attacks or spells as usual.

## Dimensional Travel workshop

```
[
  {
    "kind": "hero", "id": "rift-runner", "name": "Rift Runner",
    "stats": { "level": 19, "hitPoints": 105, "armorClass": 16,
      "speedFeet": 30, "abilityScores": { "dexterity": 20 } },
    "grants": ["#blink-steps"], "actions": ["#rift-bolt"]
  },
  { "kind": "feat", "id": "blink-steps",
    "extends": "srd:5.2.1/feat/boon-of-dimensional-travel" },
  {
    "kind": "action", "id": "rift-bolt", "name": "Rift Bolt",
    "target": { "side": "enemy", "rangeFeet": 120 },
    "resolution": { "kind": "attack", "attackType": "ranged", "attackBonus": 11,
      "damage": { "dice": { "count": 1, "sides": 8 }, "bonus": 5, "damageType": "piercing" } }
  }
]
```

After a complete Attack or Magic action, the creature can teleport up to 30 feet to an unoccupied space it can see. Extra Attack and Nick finish before this teleport. Casting a spell as a Bonus Action or Reaction does not trigger it, nor does an ordinary nonmagical action such as Dodge. Beta automatically chooses a visible destination that favors distance from enemies; it does not let an author supply a destination-selection rule.

For this benefit, Beta recognizes weapon Attack sequences and a top-level `"kind": "attack"` resolution as Attack actions. A non-spell reference, group or random table is a general action, even if it contains attack rolls. For multiple weapon attacks, use `stats.attacksPerAction` with weapon delivery. A spell cast with an Action is a Magic action; merely marking a non-spell payload `magical` does not make it one.

For an original trait, use `{ "kind": "boonOfDimensionalTravel" }`. An optional `whenEquipment` restriction controls whether it is available. The flag is permanent, has no adjustable distance or use count, and does not stack. Put any ability-score increase in the final stats. 2024 Boon of Dimensional Travel.

## Night Spirit: one grant, two benefits

Night Weaver shows a trait that provides both a Bonus Action and passive protection. Use one grant for the complete package; do not also add generic invisibility or unconditional resistances.

## Night Spirit workshop

```
[
  {
    "kind": "hero", "id": "night-weaver", "name": "Night Weaver",
    "stats": { "level": 19, "hitPoints": 90, "armorClass": 15,
      "speedFeet": 30, "darkvisionFeet": 60, "abilityScores": { "dexterity": 20 } },
    "grants": ["#shadow-thread"], "actions": ["#thread-bolt"]
  },
  { "kind": "ability", "id": "shadow-thread", "name": "Shadow Thread",
    "modifiers": [{ "kind": "boonOfNightSpirit" }] },
  {
    "kind": "action", "id": "thread-bolt", "name": "Thread Bolt",
    "target": { "side": "enemy", "rangeFeet": 60 },
    "resolution": { "kind": "attack", "attackType": "ranged", "attackBonus": 11,
      "damage": { "dice": { "count": 2, "sides": 6 }, "bonus": 5, "damageType": "necrotic" } }
  }
]
```

In Dim Light or Darkness, Night Weaver may use a Bonus Action to become Invisible. That activation does not end itself. Invisibility then lasts until the creature completes another Action, Bonus Action or Reaction; all attacks of one Attack action finish first. There is no concentration requirement or hour timer. Moving into Bright Light alone does not end the invisibility.

Independently, qualifying lighting grants Resistance to every damage type except Psychic and Radiant, whether or not Night Weaver is Invisible. Bright Light removes that protection. Being Blinded does not create darkness. The modeled Darkness spell qualifies when its active area covers the creature's whole space on the same layer; suppressed magical Darkness does not.

To use the named feat instead, replace Shadow Thread's modifier with `"extends": "srd:5.2.1/feat/boon-of-the-night-spirit"`. Both forms have the same mechanics. You may guard the modifier with `whenEquipment`; losing the last active grant immediately ends only this source of invisibility. Restoring equipment does not automatically reactivate it. Duplicate grants do not stack, and a remaining independent grant keeps the benefit available. This permanent flag accepts no amount, dice, custom duration or damage-type list. Ability-score increases belong in the final stats. 2024 Boon of the Night Spirit.

## Irresistible Offense: choose the full score

Some feats need a choice even though their ability-score increase is already included in the creature's stats. Irresistible Offense uses a selected Strength or Dexterity score for its extra damage. Declare that choice explicitly on its modifier, including when extending the SRD feat.

## Irresistible Offense workshop

```
[
  {
    "kind": "hero", "id": "wardbreaker", "name": "Wardbreaker",
    "stats": { "level": 19, "hitPoints": 130, "armorClass": 17,
      "abilityScores": { "strength": 23, "dexterity": 16 } },
    "grants": ["#overwhelming-training"], "actions": ["#fracture-blade"]
  },
  {
    "kind": "feat", "id": "overwhelming-training",
    "extends": "srd:5.2.1/feat/boon-of-irresistible-offense",
    "modifiers": [{ "kind": "irresistibleOffense", "ability": "strength" }]
  },
  {
    "kind": "action", "id": "fracture-blade", "name": "Fracture Blade",
    "activation": "action", "delivery": "weapon",
    "weapon": { "attackAbility": "strength", "opportunityAttack": true },
    "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
    "resolution": { "kind": "attack", "attackType": "melee",
      "attackBonus": 12, "criticalThreshold": 19,
      "onHit": [{ "op": "damage", "damageType": "slashing",
        "amount": { "dice": { "count": 1, "sides": 8 }, "bonus": 6 } }] }
  }
]
```

Wardbreaker ignores Resistance to all Bludgeoning, Piercing and Slashing damage it deals, including spells and other damage that does not use an attack roll. Immunity still prevents that damage. Its natural 20 adds **23** damage, the full Strength score, not the +6 ability modifier. This adds no resource or extra action and has no once-per-turn limit.

Fracture Blade also demonstrates the difference between critical range and a natural-20 benefit: a 19 rolls its normal extra critical dice, but only a 20 also gains the 23 damage. Automatic critical hits likewise need a natural 20 to receive this bonus. A spell attack can qualify too; a saving throw cannot. The bonus is not doubled again and applies once to the attack's first damage operation against its target, using that operation's damage type. Nested attacks make their own rolls; separate saving-throw damage does not inherit the bonus.

For a Dexterity-based version, change the modifier's `ability` to `dexterity`; the ability used by an individual weapon does not choose the feat's score. Keep any score increase in the final stats. An original trait can use the same modifier without `extends`. Optional `whenEquipment` restrictions work as usual, but weapon filters, custom amounts and temporary attachments are not supported. Duplicate grants do not stack; simultaneous grants must choose the same ability. Use the existing choice system to select between configured Strength and Dexterity grants. 2024 Boon of Irresistible Offense.

## Replanned weapon sequences

Use a monster or hero's top-level `attackSequences` when its Attack action requires particular weapons in particular combinations. Unlike a generic resolution sequence, the creature can move, change targets, or forgo remaining attacks between these weapon attacks. This models one Bite and one Claw in either order without allowing two Bites.

### Complementary weapon attacks

```
"actions": ["#bite", "#claw"],
"attackSequences": [
  ["#bite", "#claw"],
  ["#claw", "#bite"]
]
```

Each referenced definition must be a local `delivery: "weapon"` action already granted exactly once, unconditionally. Parameterized grants are allowed if the binding is unambiguous. Declare 1–16 sequences with 1–16 references each; duplicate sequences, missing references, generic actions, conditional weapon grants and unlisted granted weapons are rejected. Common prefixes share a state without allowing branches to exchange their remaining attacks.

`attackSequences` and explicit `stats.attacksPerAction` are mutually exclusive. Without sequences, the default remains one unrestricted weapon attack, or the stated `attacksPerAction` budget. Sequences do not repeat generic spell actions, consume extra Actions, change opportunity-attack economy, or grant spell/weapon replacements. The current field does not choose between different multiattack budgets through placement choices. The editor's Explicit weapon sequences control accepts one comma-separated sequence per line; disabling it returns to the default one-attack budget.

# 15. Add temporary bonuses and penalties

Protective Bark was a permanent +1 AC modifier given through a grant. For a blessing or curse that begins during combat and later ends, define a temporary `effect`, then use an action's `applyEffect` operation to put it on a recipient. The action is available to the caster; the modifier is active on the recipient only while the attachment lasts.

A `modifiers` array on a monster, hero, ability, feat, or ordinary mastery bundle is permanent while that definition is granted. A weapon-bound mastery capability such as Nick cannot carry this array; grant a separate ability instead. Placement-choice grants install only the selected alternative. The same array on an `effect` is temporary and requires exactly one `attachments` entry. Use `applyEffect` in an action or rule to attach it to a recipient; do not grant a temporary effect as a permanent feat.

| FIELD                                                            | TYPE                                  | REQUIRED | MEANING                                                                                                                                                                                                              |
|------------------------------------------------------------------|---------------------------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>armorClass</code>                                          | amount                                | No       | Add a signed flat adjustment to AC.                                                                                                                                                                                  |
| <code>weaponAttackBonus</code>                                   | amount + optional<br>whenWeapon       | No       | Add a signed flat bonus to weapon-delivery attack rolls only. Does not affect generic or spell attacks.                                                                                                              |
| <code>weaponDamageDieMinimum</code>                              | amount 1–100 + optional<br>whenWeapon | No       | Minimum face for the first weapon damage-dice entry. Applies to critical copies too; excludes flat damage and secondary dice. Matching floors take their maximum.                                                    |
| <code>weaponDamageRollTwice</code>                               | flag + optional whenWeapon            | No       | Once per global turn on the first eligible weapon hit, choose the higher of two whole weapon-dice packets, including critical copies. All grants share one use with native Savage Attacker. No amount or dice field. |
| <code>lightExtraAttackAbility</code>                             | flag + optional whenWeapon            | No       | Restore a positive attack ability modifier on the Light extra attack, including Nick, at most once. No extra attack or additional bonus on ordinary, opportunity or Cleave attacks. No amount or dice field.         |
| <code>initiativeProficiency</code> / <code>initiativeSwap</code> | flag; permanent only                  | No       | Add proficiency to Initiative once, or permit the automatic willing-ally Initiative exchange. See the Alert workshop. These cannot be temporary effects.                                                             |
| <code>spellSlotRecall</code>                                     | flag; permanent only                  | No       | Roll one d4 when casting with a level 1–4 slot; a matching result retains the slot and the turn's slot allowance. Does not stack. See the Spell Recall workshop in the resources chapter.                            |
| <code>combatProwess</code>                                       | flag; permanent only                  | No       | Turn a missed weapon or spell attack into a hit, once until the start of the creature's own next turn. Does not alter the natural die, grant another attack or stack.                                                |
| <code>boonOfFate</code>                                          | flag; permanent only                  | No       | Influence a completed D20 Test within 60 feet with 2d4. One shared use refreshes after rolling Initiative or a Short/Long Rest. No sight, Reaction, custom pool or stacking.                                         |
| <code>boonOfTruesight</code>                                     | flag; permanent only                  | No       | 60-foot Truesight using the shared visibility rules. Duplicates do not stack. No score increase, custom range or temporary attachments; see the workshop for Beta perception limits.                                 |
| <code>boonOfDimensionalTravel</code>                             | flag; permanent only                  | No       | Teleport up to 30 feet after a complete Attack or Magic action. The simulator selects a visible unoccupied destination; Bonus Action and Reaction spells do not trigger it. No stacking or custom range.             |
|                                                                  |                                       |          |                                                                                                                                                                                                                      |

| FIELD                                                                    | TYPE                                          | REQUIRED | MEANING                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------|-----------------------------------------------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>boonOfNightSpirit</code>                                           | flag; permanent only                          | No       | Bonus Action invisibility in Dim Light or Darkness, plus lighting-dependent Resistance except Psychic and Radiant. See the workshop for ending conditions and equipment guards. No stacking or custom duration.                          |
| <code>irresistibleOffense</code>                                         | ability: strength   dexterity; permanent only | No       | Ignore physical damage Resistance, not Immunity. A natural-20 attack adds the selected full score once to its first damage operation against its target. Other criticals and saving throws do not qualify; no score increase is applied. |
| <code>walkSpeed</code>                                                   | amount in feet                                | No       | Add a signed adjustment to walking speed, in multiples of 5 feet. Walking speed cannot become negative. This does not grant flight or swimming speed and does not directly replenish already-spent movement.                             |
| <code>savingThrowBonus</code>                                            | amount + optional ability                     | No       | Add a signed flat save adjustment. Omit ability for all saving throws, including death saves; otherwise name exactly one ability.                                                                                                        |
| <code>attackBonusDie</code> / <code>attackPenaltyDie</code>              | dice                                          | No       | Add independent bonus or penalty dice to attack rolls, including weapon and spell attacks. These do not change critical thresholds.                                                                                                      |
| <code>savingThrowBonusDie</code> / <code>savingThrowPenaltyDie</code>    | dice + optional ability                       | No       | Add independent dice to all saves or one named ability. Unfiltered entries include death saves; Constitution-only entries do not. These never modify ability checks.                                                                     |
| <code>attackAdvantage</code> / <code>attackDisadvantage</code>           | kind only                                     | No       | Add an attack-roll source. Multiple sources do not stack; Advantage and Disadvantage cancel.                                                                                                                                             |
| <code>savingThrowAdvantage</code> / <code>savingThrowDisadvantage</code> | optional ability                              | No       | Add a save-roll source, with the same cancellation rule. Omit ability for all saving throws, including death saves.                                                                                                                      |

Flat `amount` is a literal integer from -100 to 100, except `weaponDamageDieMinimum` requires 1–100. Dice use literal `count` 1–10 and `sides` 2–100. Every kind accepts `whenEquipment` for the armor/shield requirements documented under weapon delivery; the requirements check the recipient's equipment. Only the four weapon-specific kinds additionally accept `whenWeapon`. Expressions, arbitrary other filters, and additional fields are not accepted. Ability keys are strength, dexterity, constitution, intelligence, wisdom, and charisma. An attack modifier cannot take an ability filter. Changing kind resets incompatible fields but preserves equipment requirements; weapon requirements are preserved only between weapon-specific kinds.

A permanent defensive feat

```
{
  "kind": "feat", "id": "steady-footing", "name": "Steady Footing",
  "modifiers": [
    { "kind": "armorClass", "amount": 1 },
    { "kind": "savingThrowBonus", "ability": "dexterity", "amount": 2 }
  ]
}
```

Grant `#steady-footing` from a creature's `grants` list. These bonuses are explicit; the feat name does not add any native feat behavior or automatically check equipment.

### A temporary Bless-style modifier effect

```
{
  "kind": "effect", "id": "guiding-light", "name": "Guiding Light",
  "modifiers": [
    { "kind": "attackBonusDie", "dice": { "count": 1, "sides": 4 } },
    { "kind": "savingThrowBonusDie", "dice": { "count": 1, "sides": 4 } }
  ],
  "attachments": [{
    "kind": "combatant", "duration": "concentration", "maxTargetTurns": 10
  }]
}
```

### An action installs the modifier effect

```
{
  "kind": "spell", "id": "cast-guiding-light", "name": "Guiding Light",
  "level": 1, "school": "enchantment", "concentration": true,
  "target": { "side": "ally", "rangeFeet": 30, "includeSelf": true },
  "payments": [{ "id": "daily", "mode": "daily", "resource": "guiding-uses" }],
  "resolution": { "op": "applyEffect", "ref": "#guiding-light" }
}
```

The owner must declare a `guiding-uses` resource and grant `#cast-guiding-light`. This recipe intentionally targets one creature. Use `maxTargets: 3` for a bounded group; `mode: "all"` instead affects every legal recipient in range, not an implicit maximum of three.

### Guiding Chorus — bounded support

```
[
  {
    "kind": "spell", "id": "guiding-chorus", "name": "Guiding Chorus",
    "level": 1, "school": "enchantment", "components": ["verbal"], "concentration": true,
    "target": { "side": "ally", "rangeFeet": 30, "includeSelf": true, "maxTargets": 3 },
    "resource": { "ref": "lantern-oil", "cost": 1 },
    "resolution": { "op": "applyEffect", "ref": "#chorus-effect" }
  },
  {
    "kind": "effect", "id": "chorus-effect", "name": "Kindly Guidance",
    "modifiers": [
      { "kind": "attackBonusDie", "dice": { "count": 1, "sides": 4 } },
      { "kind": "savingThrowBonusDie", "dice": { "count": 1, "sides": 4 } }
    ],
    "attachments": [{ "kind": "combatant", "duration": "concentration", "maxTargetTurns": 10 }]
  }
]
```

These are the exact two definitions in Lantern Medium 0.1.3. Import that package and select `{ "guidance": "chorus" }`. Its existing oil pool supplies the cost. The original `calling` default still grants its concentration-bound summon instead. Guiding Chorus is an original verbal-only Homebrew spell, not a name-based import of native Bless.

## Attachment lifetime and ownership

Attachment `duration` accepts encounter (the default), concentration, startOfSourceNextTurn, endOfSourceNextTurn, endOfTargetNextTurn, repeatSaveEndOfTurn, or untilDamage. It never accepts instant. Optional `maxTargetTurns` is 1–600 and expires at the Nth subsequent target end-turn notification, before a repeat save. A source-next-turn duration survives its creation turn. A target-next-turn duration expires at the next end-of-target-turn notification, including the current turn if applied before that notification.

For repeated saves, supply `repeatSave: { "ability": "wisdom", "dc": 15 }`; DC is 0–50. Other durations must omit that field. A success removes this attachment only. Until-damage expiry requires positive damage to its recipient, not merely an attempted attack or absorbed packet.

`magical: true` marks a non-spell attachment magical; a spell or magical originating action makes it magical automatically. Antimagic suppresses the bonus without deleting its stored entries. Duration caps still advance during suppression, while magical repeat saves and until-damage expiry are suppressed. Nonmagical permanent grants remain active.

`persistsAfterSourceDeath` defaults to true for nonconcentration attachments. Set it false to remove the attachment during the source's normal death cleanup. Concentration always ends on source death regardless of this field. One concentration cast owns all of its modifier targets together; breaking or replacing it removes only those attachments, preserving unrelated effects and permanent grants.

**Current stacking contract:** reapplying the same compiled effect definition to one recipient replaces that attachment, including when a different caster applies it. Different definitions and independently granted permanent entries add together. User-defined stacking keys, strongest-effect selection, relational/aura/zone attachments, and consume-on-roll counters are not implemented by this attachment form. Parameters, event rules, resources, choices, and grants on a modifier attachment are rejected rather than partially installed.

Keep each definition to at most 64 modifiers. Use modest dice bonuses and avoid repeatedly creating distinct copies of the same buff: different effect definitions can stack, whereas reapplying the same definition replaces its existing attachment.

# 16. Build random abilities

Start with the one-in-three branch below. Once that is clear, try a table with several outcomes, then add repeated draws and rules about duplicates. These are the same building blocks used by Manyfaced Nightmare's changing faces.

A random ability chooses an outcome when used. Use `randomBranch` for a simple chance, or `randomTable` for several weighted or dice-mapped outcomes. Manyfaced Nightmare shows how several actions can draw from the same table.

## Random tables

- **Weighted table:** every outcome has a positive integer `weight`; the sum of all weights must be at most 255.
- **Dice table:** provide `draw.dice`; every outcome has one or more unique integer `rolls`. Multi-die sums use the real non-uniform distribution.
- Do not mix `weight` and `rolls` in one table.
- An outcome contains exactly one of `ref`, `resolve`, `operations`, or `noEffect: true`.
- Use an explicit `fallback` whenever rolls are uncovered or duplicate filtering can exhaust eligible outcomes.

## Duplicate policy and scope

| FIELD                                      | TYPE  | REQUIRED | MEANING                                                                                                                                                                                      |
|--------------------------------------------|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>allow</code>                         | mode  | No       | The same outcome may be selected repeatedly.                                                                                                                                                 |
| <code>reroll</code>                        | mode  | No       | Rejected duplicates are rolled again and preserved in replay diagnostics.                                                                                                                    |
| <code>withoutReplacement</code>            | mode  | No       | Samples from the remaining eligible distribution.                                                                                                                                            |
| <code>actionResolution / resolution</code> | scope | No       | Uniqueness lasts only for the current action resolution.                                                                                                                                     |
| <code>self.turn / turn</code>              | scope | No       | Uniqueness lasts through the current battle turn, including any off-turn actions taken during that turn. It resets when another combatant's turn starts, not only when this owner next acts. |
| <code>round</code>                         | scope | No       | Uniqueness lasts for the current round.                                                                                                                                                      |
| <code>encounter</code>                     | scope | No       | Uniqueness lasts for the encounter.                                                                                                                                                          |
| <code>rest</code>                          | scope | No       | Uniqueness survives encounter reconstruction and resets at the next short or long rest.                                                                                                      |

For turn, round, encounter and rest scopes, history belongs to the owner, referenced table and bound parameters—not to the entry action or payment name. Main actions, legendary actions and alternative payments that reference the same table therefore share the remaining outcomes. A copied table is a different table. Other owners and different parameter bindings have separate histories; integer values and damage-type sets both participate in this identity. Resolution-scoped history is confined to one resolution and does not carry into the next action.

Changing the entry action does not reset a table's shared history. Rest-scoped uniqueness lasts across consecutive encounters until a short or long rest; refreshing legendary points is not a rest.

## Target timing

`beforeDraw` chooses one target using expected utility before sampling. `afterDraw` samples first, then chooses the best target for that outcome. The `beforeEachDraw` and `afterEachDraw` variants repeat that behavior for every draw in a sequence.

## Random branches and selectors

A `randomBranch` uses `numerator` / `denominator` (aliases: `weight` / `outOf` ) and nested `then` / `else` resolutions. A random selector is an ordinary selector with `mode: "random"`.

The denominator is 1–255 and the numerator is 0 through the denominator. Defaults are 1/2. Random branches belong inside a resolution tree; rule operation arrays accept operations only. Per-outcome `when` eligibility is not implemented.

### One-in-three chance of healing the owner

```
{
  "kind": "randomBranch",
  "numerator": 1,
  "denominator": 3,
  "then": { "op": "heal", "target": "owner", "amount": 6 },
  "else": { "op": "noEffect" }
}
```

## Three distinct projections with per-draw targeting

```
{
  "kind": "action",
  "id": "changing-projections",
  "name": "Changing Projections",
  "magical": true,
  "activation": {
    "economy": "action",
    "countsAs": "multiattack"
  },
  "resolution": {
    "kind": "sequence",
    "repeat": 3,
    "steps": [
      {
        "target": {
          "timing": "beforeEachDraw",
          "selector": {
            "side": "enemy",
            "rangeFeet": 120,
            "requiresSight": true,
            "reuseTargets": false
          }
        },
        "resolve": {
          "kind": "randomTable",
          "id": "projection-table",
          "draw": {
            "dice": {
              "count": 1,
              "sides": 4
            }
          },
          "duplicates": {
            "mode": "reroll",
            "scope": "self.turn"
          },
          "fallback": {
            "op": "noEffect"
          },
          "outcomes": [
            {
              "rolls": [
                1
              ],
              "resolve": {
                "op": "damage",
                "damageType": "fire",
                "amount": 8
              }
            },
            {
              "rolls": [
                2
              ],
              "resolve": {
                "op": "applyCondition",
                "condition": "restrained",
                "duration": "endOfSourceNextTurn"
              }
            }
          ]
        }
      }
    ]
  }
}
```

```

    },
    {
      "rolls": [
        3
      ],
      "resolve": {
        "op": "push",
        "distanceFeet": 10
      }
    },
    {
      "rolls": [
        4
      ],
      "resolve": {
        "op": "damage",
        "damageType": "necrotic",
        "amount": 8
      }
    }
  ]
}
]
}
}

```

This is a complete action definition with all four outcomes inline. Give it to a creature through its actions list. It draws three projections with separate targets; each outcome can occur only once per turn. Manyfaced Nightmare's full package instead shares a referenced table between its main and Legendary Actions.

## Keep random abilities bounded

An action can make at most 16 random table or branch selections, counting repeated references and every target. This is not a limit on attack, save, or damage dice. For example, two table draws repeated eight times use 16 selections. Keep tables to at most 64 outcomes, at most four nested random constructs, and give uncertain repeats an explicit upper bound:

### Repeat expression bounded to at most eight

```

{
  "kind": "min",
  "operands": [8, { "param": "count" }]
}

```

A top-level `mode: "all"` repeats the entire resolution for every target, so any random payload is rejected unless the selector's side is `self`, which has at most one target. A table's own `all` selector draws once, then applies its selected payload to every target: this is allowed if that payload contains no further tables or random branches. Nested random payloads under such an unbounded target set are rejected. Non-random area damage remains valid, including damage dice. Use a single-target selector or place one table outside the area payload when that matches your intended mechanics; validation does not silently change the targeting or truncate execution.

## 17. Model advanced monster abilities

The remaining operations combine the concepts you have learned: a target, an attack or save, an effect, and sometimes a resource or trigger. Use the smallest operation that matches the ability. For example, Blood Drain needs one linked `lifeDrain` operation so healing and maximum-HP loss use the damage actually taken.

Operations run in array order. Most accept `target`: `owner` / `self` or `selected` / `target` / `eventTarget`. An action defaults to its selected target; an event rule defaults to its owner.

| FIELD                                 | TYPE                            | REQUIRED | MEANING                                                                                                                                                                                                                                                                                                               |
|---------------------------------------|---------------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>damage</code>                   | amount/value + damageType       | No       | Deal one typed damage packet. Use damageType inherited only in a compatible damage/attack event rule.                                                                                                                                                                                                                 |
| <code>heal</code>                     | amount/value                    | No       | Restore nonnegative hit points up to maximum, respecting healing prevention. It can revive a downed recipient when the selector permits one; it never resurrects a dead creature.                                                                                                                                     |
| <code>lifeDrain</code>                | amount/value + damageType       | No       | Deal one typed damage packet (necrotic by default), reduce the recipient's maximum HP by damage taken until long rest, and heal the owner by that amount. No configurable duration. Saves, restrictions, and action economy belong on the enclosing action.                                                           |
| <code>temporaryHitPoints</code>       | amount/value                    | No       | Grant temporary hit points through the normal engine rules.                                                                                                                                                                                                                                                           |
| <code>applyCondition</code>           | condition + duration            | No       | Apply a condition, optionally with a repeat save.                                                                                                                                                                                                                                                                     |
| <code>applyEffect</code>              | ref + optional target           | No       | Attach a local modifier effect. Declare its modifiers and lifetime on the referenced effect, not on this operation. See Typed modifiers and attachments.                                                                                                                                                              |
| <code>removeCondition</code>          | condition                       | No       | Clear all matching ordinary and save-backed condition entries, from every source. An already-clear target is unchanged.                                                                                                                                                                                               |
| <code>grapple</code>                  | escapeDC + capacity + reachFeet | No       | Create a source-owned, capacity-limited capture, optionally with dependent incapacitation or restraint. See the owned-capture example below.                                                                                                                                                                          |
| <code>push</code> / <code>pull</code> | distanceFeet or amount/value    | No       | Move the target away from or toward the source using legal geometry.                                                                                                                                                                                                                                                  |
| <code>teleport</code>                 | distanceFeet + requiresSight    | No       | Distance is 5–1,000 feet in 5-foot increments. Sight defaults to true. Chooses a legal tactical destination within that distance.                                                                                                                                                                                     |
| <code>antimagicArea</code>            | area + optional duration        | No       | Creates or refreshes one owner-attached antimagic area. Duration defaults to <code>startOfSourceNextTurn</code> , currently the only accepted duration. The selected action target sets cone/line direction, not the area center.                                                                                     |
| <code>kill</code>                     | no value                        | No       | Kill a non-dead target, including a downed or stabilized creature. Target selection still needs <code>includeDowned</code> for a zero-HP recipient. Repeating kill on an already dead target emits no second death event. Use an explicit save/condition branch to give the operation meaningful legality and threat. |
| <code>noEffect</code>                 | no value                        | No       | An explicit bounded empty branch or fallback.                                                                                                                                                                                                                                                                         |
| <code>spendResource</code>            | resource + amount/value         | No       | Atomically spend from a declared resource; insufficient uses make the operation invalid.                                                                                                                                                                                                                              |
| <code>restoreResource</code>          | resource + amount/value         | No       | Restore a declared flat uses pool up to its maximum.                                                                                                                                                                                                                                                                  |

| FIELD           | TYPE                                | REQUIRED | MEANING                                                                                                                                                                                    |
|-----------------|-------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| terminateEffect | no value                            | No       | Break the selected operation target's concentration. It does not remove an arbitrary effect by ID.                                                                                         |
| summon          | ref + count + rangeFeet + maxActive | No       | Create local monster definitions on the owner's team. Count is 1–16 per resolution; placement is within 0–1,000 feet; maxActive is 1–64, default 16, for this owner and summon definition. |
| transform       | ref + form options                  | No       | Adopt an executable local creature's body and actions while retaining real HP and identity. See Change a creature's form.                                                                  |
| revertForm      | recipient                           | No       | End the recipient's active DEF transformation and restore its original body/actions.                                                                                                       |
| createArea      | zone                                | No       | Create a persistent radius, cone or line with custom resolution callbacks. See Leave an area on the battlefield.                                                                           |

Valid damage types are bludgeoning, piercing, slashing, acid, cold, fire, force, lightning, necrotic, poison, psychic, radiant, and thunder. `inherited` is additionally accepted only where the triggering event supplies a damage type.

**Blood Drain: linked damage, maximum HP, and healing**

Use `lifeDrain`, not independent damage and heal rolls. Resistance, immunity, damage absorption, and defensive reactions affect the damage first. The resulting damage taken determines both maximum-HP loss and owner healing. Damage absorbed by temporary HP and damage beyond remaining current HP still count. Maximum HP cannot fall below zero; reaching zero kills the recipient. Current HP is clamped to the reduced maximum. Healing is capped at the owner's maximum and respects healing prevention; it cannot resurrect a dead owner.

The maximum-HP reduction survives encounter-chain transitions and short rests, and recovers on a successful long rest. A curse that prevents rest benefits still blocks recovery. This tracks only the recoverable drain loss, without removing unrelated permanent or curse-based maximum-HP reductions. Each `lifeDrain` is a separate packet, never combined with neighboring damage operations. Replays record the target's changed maximum HP and the owner's healing. On an attack's critical hit, its damage dice double; a saving throw such as Blood Drain does not critically hit.

Blood Drain — weaker Blood Druid version

```
{
  "kind": "action", "id": "blood-drain", "name": "Blood Drain",
  "activation": "bonusAction", "magical": true,
  "target": {
    "side": "enemy", "rangeFeet": 30, "requiresSight": true,
    "excludeCreatureTypes": ["construct", "undead"]
  },
  "resolution": {
    "kind": "savingThrow", "ability": "constitution", "dc": 16,
    "onFailure": {
      "op": "lifeDrain", "damageType": "necrotic",
      "amount": { "dice": { "count": 3, "sides": 6 } }
    }
  }
}
```

The Blood Druid package uses DC 16 and 3d6 for the weaker drain. To make a stronger version, increase the DC or dice count. A successful save does nothing: do not add `halfOnSave`. Blood Drain is magical but is not a spell. CR 8 is provisional, not a verified balance rating. Withering Pollen adds its own limited-use poison and healing-prevention payload; Funeral Seed is attached only to Blood Drain's causal kill result.

## Own-damage death, disintegration, and Funeral Seed

Do not place an unconditional `kill` after damage to represent “dies if this damage reaches zero HP.” Use `onZeroHitPoints` on that specific damage or lifeDrain operation. `kill` causes immediate death only if this packet takes positive HP to zero and the victim is still at zero. `disintegrate` also removes that creature's remains from corpse-based revival/animation lookup. A different callback killing the victim does not qualify.

### Disintegration attached to one packet

```
{
  "op": "damage",
  "amount": { "dice": { "count": 7, "sides": 8 } },
  "damageType": "force",
  "onZeroHitPoints": "disintegrate"
}
```

This models creature damage and destroyed remains, not equipment destruction, arbitrary objects, force constructs, or every tabletop disintegration interaction. Native survival replacements and healing reactions still matter. A consequence-bearing packet is kept separate from neighboring plain damage operations; an earlier or later packet cannot borrow its consequence.

### Blood Drain with the original Funeral Seed trait

```
{
  "op": "lifeDrain",
  "amount": { "dice": { "count": 3, "sides": 6 } },
  "damageType": "necrotic",
  "onKillSummon": {
    "ref": "#gloom-sprout",
    "creatureTypes": ["humanoid"],
    "maximumSize": "medium",
    "maxActive": 1,
    "rangeFeet": 5
  }
}
```

`onKillSummon` is allowed only on damage/lifeDrain. Supply a local monster reference and a nonempty creature-type list. Optional `maximumSize` limits the victim, not the minion. `maxActive` is 1–16 (default 1), counted among living summons of this definition belonging to this owner. `rangeFeet` is 0–1,000 in five-foot increments (default 5), measured around the victim's former position. The owner must remain alive; the victim must be a qualifying enemy killed by this operation, including its own zero-maximum-HP drain consequence.

Placement happens after death cleanup frees the victim's square. Successful creation consumes only this victim's remains, places the minion on the owner's team, and lets the minion take its own turns. An occupied placement, a full cap, an

ineligible victim, or an unrelated kill creates nothing. A supporting minion is an ordinary authored monster, not a hidden native preset. General corpse raising and indefinite campaign persistence are separate mechanics.

## Owned captures and dependent conditions

A two-arm capture rider

```
{
  "op": "grapple",
  "escapeDC": 16,
  "capacity": 2,
  "reachFeet": 10,
  "maximumSize": "medium",
  "dependentCondition": "incapacitated"
}
```

escapeDC is required, 0–50. capacity is 1–16 (default 1); reachFeet is 5–1,000 in five-foot increments (default 5). maximumSize is optional. dependentCondition is omitted, incapacitated, or restrained. A grapple does not take a generic duration, repeatSave, or condition field: its lifetime belongs to the hold.

The operation requires a living, non-incapacitated owner, an eligible target within reach, and a free capture slot. It does not impose another initial save; put it inside an attack's onHit branch or a savingThrow failure branch. Immunity to Grappled prevents the hold; immunity to the dependent rider prevents that rider without discarding an otherwise legal hold. Capacity counts this owner's distinct captives across its capture operations, not a separate pool per action.

Escape uses the native Action/Strength (Athletics) or Dexterity (Acrobatics) check. Escape, the captor's incapacitation/death, or actual separation beyond the maintained reach ends that source's hold and dependent rider. Multiple captors retain independent links. Planner-only hypothetical movement does not release a real hold. This primitive does not automatically drag a captive along with the owner. A generic cure removes a matching composite condition as a whole, so curing its dependent rider also releases that hold.

Brain extraction requires this octopus's incapacitating hold

```
"target": {
  "side": "enemy", "rangeFeet": 5,
  "requiresOwnedGrapple": true,
  "requiresCondition": "incapacitated",
  "maximumSize": "medium"
}
```

Add creature-type exclusions for the intended anatomy; the Psy-Octopus package permits humanoid victims. Another creature's capture, or an unrelated incapacitating spell, cannot satisfy the owned dependent-condition requirement. The follow-up attack uses its own damage-bound death consequence.

## Confusing Display and genuine confused turns

**confused** follows the turn behavior of the 2024 Confusion spell. While affected, the creature cannot take bonus actions or reactions. At its turn's start, one d10 selects the behavior below; overlapping sources do not roll separate behavior tables.

| FIELD | TYPE                 | REQUIRED | MEANING                                                                                                                                                                                                                   |
|-------|----------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | Disordered movement  | No       | Choose a random cardinal direction, move as far as legal movement permits, and take no action. Ordinary terrain, obstacles, hazards, and opportunity attacks still apply.                                                 |
| 2-6   | Lost turn            | No       | No movement or action.                                                                                                                                                                                                    |
| 7-8   | Misdirected strike   | No       | No movement; make one melee attack against a randomly selected creature within reach, including allies. If there is no eligible creature/attack, take no action. Extra Attack, Multiattack, and Haste do not add attacks. |
| 9-10  | Normal action choice | No       | Choose movement and action normally. The condition's bonus-action and reaction prohibition still applies.                                                                                                                 |

A visual, repeat-save confusion ability

```
{
  "kind": "action", "id": "confusing-display", "name": "Confusing Display",
  "activation": "bonusAction", "magical": true,
  "target": { "side": "enemy", "rangeFeet": 30,
    "requiresSight": true, "requiresTargetSight": true,
    "excludesCondition": "confused" },
  "resolution": { "kind": "savingThrow", "ability": "wisdom", "dc": 16,
    "onFailure": { "op": "applyCondition", "condition": "confused",
      "duration": "repeatSaveEndOfTurn", "maxTargetTurns": 10,
      "repeatSave": { "ability": "wisdom", "dc": 16 } }
  }
}
```

This is the octopus's original single-target magical display, not a claim that it casts the complete area/concentration Confusion spell. Each source retains its own save/duration; magical suppression suspends the active condition while duration still advances. Curing or expiring it restores normal bonus-action/reaction eligibility, subject to other conditions and already-spent resources.

Named 2024 combat spells

Blood Druid explicitly defines Thorn Whip, Chill Touch, Poison Spray, and Blight as local spells. Their names do not import their mechanics. The example writes three-die cantrip damage and two fourth-level Blight uses per long rest; these values do not automatically scale with the monster's statistics.

Thorn Whip declares Verbal, Somatic, and Material components. The other three declare Verbal and Somatic. The spells also declare their schools: Transmutation for Thorn Whip and Necromancy for the others. Components describe casting restrictions, not an inventory of material items.

| FIELD       | TYPE                  | REQUIRED | MEANING                                                                                                                                                                                                                                                                              |
|-------------|-----------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Thorn Whip  | melee attack, 30 feet | No       | 3d6 piercing, then a pull of up to 10 feet limited to Large or smaller creatures. The size limit belongs on the pull operation, so larger creatures can still take damage. This example always attempts the full legal pull; change distanceFeet to 0 or 5 for a shorter fixed pull. |
| Chill Touch | melee attack, 5 feet  | No       | 3d10 necrotic and healing prevention until endOfSourceNextTurn, not the target's turn.                                                                                                                                                                                               |

| FIELD        | TYPE                             | REQUIRED | MEANING                                                                                                                                                                                |
|--------------|----------------------------------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Poison Spray | ranged attack, 30 feet           | No       | 3d12 poison. Uses an attack roll, not the legacy Constitution save.                                                                                                                    |
| Blight       | Constitution save, 30-foot sight | No       | 8d8 necrotic, half on success; autoFailCreatureTypes: [plant]. Plant targets take rolled damage, not maximum damage. Nonmagical plant objects are outside this combatant-only example. |

See the 2024 spell reference, Blight, and Thorn Whip (Player’s Handbook access may be required). Spells use normal magical suppression; range and sight restrictions are explicit. There is no Wild Shape or out-of-combat spell behavior hidden in this package.

## Antimagic field with a switchable shape

An owner-centered antimagic field

```
{
  "kind": "action", "id": "antimagic-field", "name": "Antimagic Field",
  "activation": "bonusAction",
  "target": { "side": "enemy", "rangeFeet": 150 },
  "resolution": {
    "op": "antimagicArea",
    "area": { "shape": "radius", "sizeFeet": 30 },
    "duration": "startOfSourceNextTurn"
  }
}
```

Change only `shape` from `radius` to `cone` to make this a directional field. Cone/line direction is chosen toward the selected target and stays fixed while the field moves with its owner. A self selector has no aiming target and therefore points toward positive map Y. Do not mark the field-creation action itself magical: the suppression source must not suppress its own operation.

**A radius suppresses its owner too.** It suppresses spells and abilities marked magical inside the footprint and magic paths crossing it, including Hushbell Colossus’s magical chimes. Cone/line fields exclude the source’s footprint. Mark every supernatural ray action, including its granted entry action, `magical: true`; referencing a payload does not inherit the referenced action’s metadata.

Suppression is temporary, not dispelling: tracked magical conditions remain stored and become active again on leaving the area or when the source’s next turn begins. Repeated activation refreshes the same owner’s field; it does not stack copies. This specialized suppression field still has a fixed duration and owner origin. Custom damaging, healing, and condition-bearing areas use `createArea`; that operation does not turn a zone into antimagic. Equipment disenchantment and suppressed summon disappearance/reappearance are not modeled.

**Turn-start creation:** a field created by a `turn.start` rule survives that same start-of-turn effect tick. It expires at the next source turn, not immediately after creation and not at another creature’s turn. A rule that creates a field every turn refreshes it at each source start. The same creation-turn protection applies to conditions with `startOfSourceNextTurn`.

## 18. Change a creature's form

A transformation changes the creature already in the encounter; it does not summon a replacement. Start with a second monster or hero definition for the body, then reference it from `transform`. Its statistics and grants describe what the new body can do.

### Transformation workshop

```
[
  {
    "kind": "hero", "id": "briar-keeper", "name": "Briar Keeper",
    "stats": { "level": 5, "hitPoints": 42, "armorClass": 13,
      "abilityScores": { "wisdom": 18 } },
    "resources": [{ "id": "shades", "maximum": 2, "refresh": "shortRest" }],
    "actions": ["#assume-shell"]
  },
  {
    "kind": "action", "id": "assume-shell", "name": "Assume Briar Shell",
    "activation": "bonusAction", "target": { "side": "self" },
    "resource": { "ref": "shades", "cost": 1 },
    "resolution": { "op": "transform", "ref": "#briar-shell",
      "durationRounds": 10, "retainMentalAbilities": true,
      "temporaryHitPoints": 10 }
  },
  {
    "kind": "monster", "id": "briar-shell", "name": "Briar Shell",
    "stats": { "hitPoints": 60, "armorClass": 17, "size": "large",
      "speedFeet": 40, "abilityScores": { "strength": 18, "constitution": 16 } },
    "actions": ["#briar-slam", "#return-to-self"]
  },
  {
    "kind": "action", "id": "briar-slam", "name": "Briar Slam",
    "target": { "side": "enemy", "rangeFeet": 5 },
    "resolution": { "kind": "attack", "attackBonus": 7,
      "damage": { "dice": { "count": 2, "sides": 6 }, "bonus": 4,
        "damageType": "bludgeoning" } }
  },
  {
    "kind": "action", "id": "return-to-self", "name": "Return to Self",
    "activation": "freeAction", "target": { "side": "self" },
    "resolution": { "op": "revertForm" }
  }
]
```

Put these definitions inside your package envelope. Assuming the shell grants 10 temporary HP and changes AC to 17, but the Keeper retains its own 42 maximum HP. Damage beyond the temporary HP reduces real HP; reverting never restores that damage. The shell's printed 60 HP is not a second life bar.

### What carries over?

The creature keeps its identity, team, initiative position, real HP, Hit Dice, creature type, conditions, concentration, and actions/movement already spent. Original passive class/feat features and rules remain; a form is not a switch to a different player class. The new body supplies base AC, movement modes, size, physical ability scores and saves, listed

skills/senses, equipment facts, and base defenses. Its actions, resources, rules and modifiers are available while shaped. Form choices use their declared defaults.

Intelligence, Wisdom, Charisma and their saves stay with the original creature by default. Original actions are replaced by default. Keeping them as alternatives does not permit mixing two full Multiattack sequences or replenish an already-started Attack action.

| FIELD                                      | TYPE              | REQUIRED | MEANING                                                                                                                                     |
|--------------------------------------------|-------------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ref</code>                           | local #reference  | Yes      | An executable monster or hero definition; native-only catalog records are not body templates.                                               |
| <code>target</code>                        | selected or owner | No       | Selected creature by default. The action can select self, an ally, or an enemy. Wrap transform in a saving throw if resistance is intended. |
| <code>durationRounds</code>                | integer 1–1000    | No       | Defaults to 10. Reverts at the start of the Nth subsequent target turn; creation does not consume a turn.                                   |
| <code>retainMentalAbilities</code>         | boolean           | No       | Defaults to true. False adopts the form's Intelligence, Wisdom, Charisma and corresponding saves.                                           |
| <code>retainActions</code>                 | boolean           | No       | Defaults to false. True keeps original actions alongside the form's alternatives.                                                           |
| <code>temporaryHitPoints</code>            | expression        | No       | Defaults to zero. The greater of existing and new temporary HP applies; pools do not add together.                                          |
| <code>endWhenTemporaryHitPointsLost</code> | boolean           | No       | Defaults to false. True reverts when no temporary HP remain; supply a <code>temporaryHitPoints</code> expression.                           |
| <code>clearTemporaryHitPointsOnEnd</code>  | boolean           | No       | Defaults to false. True removes this form's accepted grant on reversion, without removing a later replacement grant.                        |

### Reversion and practical boundaries

`revertForm` restores the original body. Death and duration expiry also revert. A concentration action ends its forms when concentration ends. Temporary HP normally survive reversion unless you opt into clearing them. Original resources stay spent, and returning to a previously used form does not refill its resources.

A free-action transformation must spend a resource on every activation; use an Action or Bonus Action for an at-will transformation. An unpaid free-action reversion uses only `revertForm` and a single Self target in Best mode, as above. Group, random or conditional reversions need an action budget or resource payment.

A larger body needs space. Reversion from a smaller form uses the nearest available space that fits the original body when necessary. Leave room for the original body somewhere on the map: if no legal space exists, the simulation reports a blocked reversion instead of overlapping creatures or walls. DEF forms cannot be layered over native Wild Shape, and recursive form/summon chains are rejected. Replacing a DEF form ends the old one first. This is a configurable homebrew transformation, not every provision of Polymorph or Wild Shape: inventory melding, changed languages and mental role-playing restrictions are not inferred.

A form created by a magical action is temporarily suppressed inside antimagic. Its duration keeps running, its original body/actions return while suppressed, and leaving the field restores the form if it fits. Suppression does not grant temporary HP again or refill resources.

## 19. Leave an area on the battlefield

`target.area` resolves a burst now. `createArea` leaves a zone behind, with geometry, duration and callbacks. A **callback** names the resolution to run when a particular event happens to an affected creature.

### Persistent area workshop

```
[
  {
    "kind": "hero", "id": "ash-gardener", "name": "Ash Gardener",
    "stats": { "level": 5, "hitPoints": 40, "armorClass": 14 },
    "actions": ["#cinder-garden"]
  },
  {
    "kind": "action", "id": "cinder-garden", "name": "Cinder Garden",
    "magical": true, "concentration": true, "target": { "side": "self" },
    "resolution": { "op": "createArea", "zone": {
      "area": { "shape": "radius", "sizeFeet": 10 },
      "anchor": "point", "rangeFeet": 60, "durationRounds": 3,
      "affects": { "side": "enemy" }, "oncePerTurn": true,
      "difficultTerrain": true, "onEnter": "#cinder-burn",
      "onTurnStart": "#cinder-burn"
    } }
  },
  {
    "kind": "action", "id": "cinder-burn", "name": "Cinder Burn",
    "resolution": { "kind": "savingThrow", "ability": "dexterity", "dc": 14,
      "halfOnSave": true, "onFailure": {
        "op": "damage", "amount": { "dice": { "count": 2, "sides": 6 } },
        "damageType": "fire"
      } }
  }
]
```

The Garden is created once, up to 60 feet away. Enemies in its 10-foot radius treat the area as difficult terrain and resolve Cinder Burn when entering or starting a turn there. Breaking concentration removes it. The Gardener need not recast each round.

### Separate placement from recipients

The creating action uses a Self target, with `createArea` as its only top-level operation. This prevents creating one area per target. The simulator chooses placement and heading during combat. The zone's `affects` filter selects occupants anew whenever a trigger occurs.

A free-action area must spend a resource on every activation, with no At-will payment alternative. An at-will area can instead use an Action or Bonus Action.

A callback executes only its referenced action/spell's `resolution`: no second action, casting payment or top-level target selection. The creating action supplies the caster, magical/concentration status and cast level. Use literals or owner/target statistics; independent callback parameters are unsupported. Do not grant the callback definition to the creature unless it should also be selectable as a separate action.

| FIELD                                 | TYPE                   | REQUIRED | MEANING                                                                                                                                                                                                                                                         |
|---------------------------------------|------------------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>area</code>                     | shape + sizeFeet       | Yes      | radius, cone or line. A line also accepts widthFeet. Uses the same two-dimensional geometry as instantaneous areas.                                                                                                                                             |
| <code>anchor</code>                   | point or owner         | No       | Defaults to point, a stationary zone. owner follows the source; a cone/line keeps its heading until recast.                                                                                                                                                     |
| <code>rangeFeet</code>                | 0–1000, multiples of 5 | No       | Point placement defaults to 60 feet and requires sight. An owner anchor has range zero.                                                                                                                                                                         |
| <code>durationRounds</code>           | integer 1–1000         | No       | Defaults to 3, expiring at the source's Nth subsequent turn start. An independently surviving zone also expires after that many combat rounds if its source dies.                                                                                               |
| <code>affects</code>                  | occupant filter        | No       | Defaults to any side including owner. Accepts side, includeSelf, includeDowned, maximumSize, requiresCondition, excludesCondition, excludeCreatureTypes, requiresSight and requiresTargetSight. All eligible occupants are considered; geometry supplies range. |
| <code>oncePerTurn</code>              | boolean                | No       | Defaults to true: one callback per creature per global turn, shared across all trigger types in this zone. Another creature's turn starts a new allowance.                                                                                                      |
| <code>difficultTerrain</code>         | boolean                | No       | Defaults to false. Applies to eligible occupants. Terrain alone is allowed without callbacks.                                                                                                                                                                   |
| <code>maxActive</code>                | integer 1–16           | No       | Defaults to 1 per source and creating action definition. At the cap, a new zone replaces the oldest.                                                                                                                                                            |
| <code>persistsAfterSourceDeath</code> | boolean                | No       | Defaults to false. Only stationary, non-concentration zones may opt in.                                                                                                                                                                                         |

| FIELD                     | TYPE             | REQUIRED | MEANING                                                                                                       |
|---------------------------|------------------|----------|---------------------------------------------------------------------------------------------------------------|
| <code>onCreate</code>     | local #reference | No       | For each eligible occupant when the area appears.                                                             |
| <code>onEnter</code>      | local #reference | No       | On entry, including occupants covered when the area appears and creatures swept into an owner-following area. |
| <code>onMoveWithin</code> | local #reference | No       | Movement that keeps a creature inside. With oncePerTurn false, movement repeatedly triggers the payload.      |
| <code>onExit</code>       | local #reference | No       | Movement out, including movement of an owner-following area.                                                  |
| <code>onTurnStart</code>  | local #reference | No       | An affected creature starts its turn while the zone is active.                                                |
| <code>onTurnEnd</code>    | local #reference | No       | An affected creature ends its turn while the zone is active.                                                  |

**Timing matters:** creation checks `onCreate` before `onEnter`. With the default shared gate, only the first configured callback runs for that creature that turn. The gate also covers leaving and turn boundaries. Set `oncePerTurn: false` when separate entry and exit payloads must both run, such as applying a condition on entry and removing it on exit.

Callbacks can use saves, attacks, damage, healing, conditions, modifiers, random tables, movement and other executable resolution operations. Conditions and modifiers retain their own declared lifetimes: leaving does not automatically remove them, and removing a zone does not manufacture exit events. Callbacks cannot create another persistent area. Filters and line of effect are rechecked on each callback. Overlapping zones have independent trigger allowances.

**Try it:** use `anchor: "owner"` with `rangeFeet: 0` for an aura. Change `area.shape` to `cone` for a directional aura, or to `line` with `widthFeet: 5` for a narrow hazard. For a healing sanctuary, choose allied occupants and replace the

callback payload with `{ "op": "heal", "amount": 5 }`.

## 20. What to expect in Beta

The supported fields in this guide create mechanics; names, flavor text, and extra JSON properties do not. Put descriptive details in `metadata`, and do not rely on an unrecognized field to change combat. A valid file still needs a short trial encounter to check that it behaves as intended.

Important boundaries when planning a creature: automatic level progression and a full inventory are not provided; only the listed weapon masteries and modifier forms are portable. Lair actions, out-of-combat actions, arbitrary scripts, remote dependencies, and global area-enter/leave rules are not supported. Use the dedicated transformation and persistent-area operations for the behaviors described above. A built-in SRD ability being present in Encounterra does not by itself make it reusable in DEF.

These examples demonstrate combat mechanics, not official creature statistics or verified challenge ratings. A broad concept such as Wild Shape, object telekinesis, or permanent campaign resurrection cannot be created by changing an action's name. If a needed mechanic is not covered here, treat it as unsupported rather than silently substituting another effect.

### Keep your packages editable

Keep an exported backup while the format is in Beta. Use `"schemaVersion": "0.1"` for new packages; numeric `0.1` is not the same value. Legacy integer `1` packages remain readable. To adopt the current spelling, make an edited version of that package instead of assuming an old saved encounter has changed.

Your `package.version` is separate from the format version. Use a new package UUID for an independent creation based on an example; keep your UUID when revising your own package. Increase your package version for a new release. Previously saved encounters retain their included version until you explicitly update them.

# 21. Ten complete worked examples

These six monsters and four heroes bring the lessons together. Each entry includes a reading guide, one change to try, and the entire importable JSON package. Nothing in the printed code is replaced by an ellipsis. Supporting minions are included in their owner’s file and are not extra featured examples.

Choose an example by the mechanic you want to learn. Download its JSON for editing, give your independent creation a new package UUID, and keep the local references consistent. The examples are original teaching builds with authored mechanics, not automatic implementations of player classes or certified challenge ratings.

| EXAMPLE                     | ROLE                                     | WHAT IT TEACHES                                                                                                                                                                      |
|-----------------------------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Manyfaced Nightmare monster | A walking cloth-and-porcelain controller | Random projections with turn-shared duplicate rerolls; post-draw tactical targeting; charm, fear, sleep and paralysis; lighting-dependent traits                                     |
| Psy-Octopus monster         | An amphibious mnemonic hunter            | Walking and swimming; visual confusion; two owned captures; brain extraction; magic-save advantage                                                                                   |
| Umbral Dragon monster       | A mobile dream-eating dragon             | Compound bite and claws; flight; necrotic cone/line and psychic stun cone; shared recharge                                                                                           |
| Blood Druid monster         | A gardener of blighted groves            | 2024 combat spells; Blood Drain; a custom persistent Blighted Grove; poison/healing prevention; causal Funeral Seed and Gloam Sprout                                                 |
| Hushbell Colossus monster   | A walking stone-and-brass sanctuary      | Radius/cone antimagic; physical hammer; disintegration; death effect; legendary budgets and finite save replacement                                                                  |
| Marbleweaver Moth monster   | An airborne quarry trapper               | Binding; narrow slowing; staged petrification; ordinary flight                                                                                                                       |
| Dawnward Sentinel hero      | A shield-bearing protector               | Weapon delivery and opportunity attacks; critical hits on 19–20; optional once-per-turn weapon dice selection; parameterized grants; automatic defense and multi-unit resource costs |
| Emberbound Adept hero       | A twofold elemental caster               | Fire/cold choices; nested parameter forwarding; inheritance; spell attacks; area saves and alternative payments                                                                      |
| Cinderroot Warden hero      | A living-wood guardian                   | Briar Shell transformation and reversion; fire-inhibited regeneration; typed event history; predicates and limits; burrowing; arithmetic; short/long rests                           |
| Lantern Medium hero         | A guide for kindly spirits               | Concentration; bounded summon or three-target modifier chorus; ally healing; temporary HP; cure; self teleport                                                                       |

## 1. Manyfaced Nightmare

Download the complete Manyfaced Nightmare package

**Start with:** `changing-faces`, `nightmare-faces`, `passing-face`, `dreaming-face`.

**How it fits together:** Begin with Changing Faces, which repeats a reference to the face table. Passing Face uses the same table through a Legendary Action. Each face supplies a different magical projection. Like a beholder's Eye Rays, the table rolls first and rerolls a face already used during the same turn; after each committed face roll, the simulator tactically picks its best eligible target.

**What to look for:** Read one outcome at a time: charm can break on damage, sleep can end when an ally wakes the victim, and paralysis has repeat saves and a separate turn cap. The creature's light-dependent statistics use the actual illumination around it; darkvision does not change the light level.

**Try changing it:** Change Changing Faces from two draws to three. Then change the table's uniqueness scope from turn to round and watch when a face becomes available again. Keep both actions referencing the same table if they should share used outcomes. Compare reroll with withoutReplacement: both exclude used faces, but reroll preserves rejected duplicate rolls in replay diagnostics. The explicit fallback handles an exhausted table.

### Complete package: Manyfaced Nightmare

```
{
  "format": "encounterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "f1c0826b-f8a9-40d5-9fc4-30b79b460614",
    "name": "Manyfaced Nightmare – Walking Dream-Terror",
    "version": "0.1.1"
  },
  "requires": {
    "ruleset": "dnd-2024",
    "srdCatalog": "5.2.1-encounterra.1"
  },
  "definitions": [
    {
      "kind": "monster",
      "id": "manyfaced-nightmare",
      "name": "Manyfaced Nightmare",
      "stats": {
        "challengeRating": "9",
        "hitPoints": 143,
        "armorClass": 16,
        "speedFeet": 30,
        "size": "large",
        "creatureType": "aberration",
        "darkvisionFeet": 120,
        "difficultyClass": 16,
        "legendaryActionUses": 2,
        "abilityScores": {
          "strength": 16,
          "dexterity": 15,
          "constitution": 18,
          "intelligence": 15,
          "wisdom": 14,
          "charisma": 19
        }
      }
    }
  ]
}
```

```

    },
    "savingThrows": {
      "strength": 3,
      "dexterity": 5,
      "constitution": 7,
      "intelligence": 2,
      "wisdom": 5,
      "charisma": 7
    },
    "dimLightResistances": [
      "bludgeoning",
      "piercing",
      "slashing",
      "acid",
      "cold",
      "fire",
      "lightning",
      "thunder"
    ],
    "brightLightSensitivity": true,
    "conditionImmunities": [
      "frightened"
    ]
  },
  "actions": [
    "#cloth-talons",
    "#changing-faces",
    "#passing-face"
  ]
},
{
  "kind": "action",
  "id": "cloth-talons",
  "name": "Cloth Talons",
  "target": {
    "side": "enemy",
    "rangeFeet": 10
  },
  "resolution": {
    "kind": "attack",
    "attackType": "melee",
    "attackBonus": 7,
    "onHit": {
      "op": "damage",
      "amount": {
        "dice": {
          "count": 3,
          "sides": 8
        },
        "bonus": 3
      },
      "damageType": "slashing"
    }
  }
},
{
  "kind": "action",
  "id": "changing-faces",
  "name": "Changing Faces",
  "target": {
    "side": "enemy",
    "rangeFeet": 120
  },

```

```

    "resolution": {
      "kind": "sequence",
      "repeat": 2,
      "steps": [
        {
          "kind": "reference",
          "ref": "#nightmare-faces"
        }
      ]
    },
    "magical": true,
    "activation": {
      "economy": "action",
      "countsAs": "multiattack"
    }
  },
  {
    "kind": "action",
    "id": "passing-face",
    "name": "Passing Face",
    "target": {
      "side": "enemy",
      "rangeFeet": 120
    },
    "resolution": {
      "kind": "reference",
      "ref": "#nightmare-faces"
    },
    "magical": true,
    "activation": "legendaryAction",
    "legendaryCost": 1
  },
  {
    "kind": "action",
    "id": "nightmare-faces",
    "name": "Nightmare Faces",
    "target": {
      "side": "enemy",
      "rangeFeet": 120
    },
    "resolution": {
      "kind": "randomTable",
      "id": "nightmare-faces",
      "target": {
        "timing": "afterEachDraw",
        "selector": {
          "side": "enemy",
          "rangeFeet": 120,
          "mode": "best",
          "requiresSight": true,
          "reuseTargets": true,
          "excludeCreatureTypes": [
            "construct",
            "undead"
          ]
        }
      }
    },
    "draw": {
      "dice": {
        "count": 1,
        "sides": 4
      }
    }
  },

```

```

    "duplicates": {
      "mode": "reroll",
      "scope": "turn"
    },
    "fallback": {
      "op": "noEffect"
    },
    "outcomes": [
      {
        "rolls": [
          1
        ],
        "ref": "#beguiling-face"
      },
      {
        "rolls": [
          2
        ],
        "ref": "#dread-face"
      },
      {
        "rolls": [
          3
        ],
        "ref": "#dreaming-face"
      },
      {
        "rolls": [
          4
        ],
        "ref": "#frozen-face"
      }
    ]
  },
  "magical": true
},
{
  "kind": "action",
  "id": "beguiling-face",
  "name": "Beguiling Face",
  "target": {
    "side": "enemy",
    "rangeFeet": 120
  },
  "resolution": {
    "kind": "savingThrow",
    "ability": "wisdom",
    "dc": 16,
    "onFailure": [
      {
        "op": "damage",
        "amount": {
          "dice": {
            "count": 3,
            "sides": 8
          }
        },
        "damageType": "psychic"
      }
    ],
    {
      "op": "applyCondition",
      "condition": "charmed",
      "duration": "untilDamage",

```

```

        "maxTargetTurns": 10
    }
]
},
"magical": true
},
{
    "kind": "action",
    "id": "dread-face",
    "name": "Dread Face",
    "target": {
        "side": "enemy",
        "rangeFeet": 120
    },
    "resolution": {
        "kind": "savingThrow",
        "ability": "wisdom",
        "dc": 16,
        "onFailure": [
            {
                "op": "damage",
                "amount": {
                    "dice": {
                        "count": 3,
                        "sides": 8
                    }
                },
                "damageType": "psychic"
            },
            {
                "op": "applyCondition",
                "condition": "frightened",
                "duration": "endOfTargetNextTurn"
            }
        ]
    },
    "magical": true
},
{
    "kind": "action",
    "id": "dreaming-face",
    "name": "Dreaming Face",
    "target": {
        "side": "enemy",
        "rangeFeet": 120
    },
    "resolution": {
        "kind": "savingThrow",
        "ability": "wisdom",
        "dc": 16,
        "onFailure": {
            "op": "applyCondition",
            "condition": "unconscious",
            "duration": "untilDamage",
            "maxTargetTurns": 10,
            "canBeShakenAwake": true
        }
    },
    "magical": true
},
{
    "kind": "action",
    "id": "frozen-face",

```

```

    "name": "Frozen Face",
    "target": {
      "side": "enemy",
      "rangeFeet": 120
    },
    "resolution": {
      "kind": "savingThrow",
      "ability": "constitution",
      "dc": 16,
      "onFailure": {
        "op": "applyCondition",
        "condition": "paralyzed",
        "duration": "repeatSaveEndOfTurn",
        "repeatSave": {
          "ability": "constitution",
          "dc": 16
        },
        "maxTargetTurns": 10
      }
    },
    "magical": true
  }
]
}

```

## 2. Psy-Octopus

Download the complete Psy-Octopus package

**Start with:** `confusing-display`, `seizing-arms`, `devour-brain`.

**How it fits together:** Confusing Display checks whether both creatures can see one another and applies Confused. Seizing Arms creates up to two captures belonging to this octopus, each with dependent incapacitation. Devour Brain then requires an incapacitated target held by this same owner; another creature's grapple is not enough.

**What to look for:** Read the capture and its follow-up together. Escaping or breaking the hold removes the dependent condition. Confused has its own turn behavior, including lost actions and misdirected movement or attacks; it is not a differently named Stunned condition.

**Try changing it:** Reduce capture capacity to one and compare the available targets for Seizing Arms and Devour Brain. Try a target that cannot see the octopus to understand requiresTargetSight. Walking and swimming are explicitly supplied; thought communication is flavor, not a hidden combat mechanic.

### Complete package: Psy-Octopus

```
{
  "format": "encounterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "b5f380e8-1438-4fc5-9f43-2a6e3f780ec0",
    "name": "Psy-Octopus – Mnemonic Hunter",
    "version": "0.1.0"
  },
  "requires": {
    "ruleset": "dnd-2024",
    "srdCatalog": "5.2.1-encounterra.1"
  },
  "definitions": [
    {
      "kind": "monster",
      "id": "psy-octopus",
      "name": "Psy-Octopus",
      "stats": {
        "challengeRating": "7",
        "hitPoints": 120,
        "armorClass": 16,
        "speedFeet": 30,
        "swimSpeedFeet": 30,
        "size": "large",
        "creatureType": "aberration",
        "darkvisionFeet": 60,
        "difficultyClass": 16,
        "magicResistance": true,
        "abilityScores": {
          "strength": 18,
          "dexterity": 14,
          "constitution": 17,
          "intelligence": 18,
          "wisdom": 16,
          "charisma": 10
        },
        "savingThrows": {
```

```

        "strength": 4,
        "dexterity": 2,
        "constitution": 3,
        "intelligence": 7,
        "wisdom": 6,
        "charisma": 0
    }
},
"actions": [
    "#seizing-arms",
    "#devour-brain",
    "#confusing-display"
]
},
{
    "kind": "action",
    "id": "seizing-arms",
    "name": "Seizing Arms",
    "target": {
        "side": "enemy",
        "rangeFeet": 10
    },
    "resolution": {
        "kind": "attack",
        "attackType": "melee",
        "attackBonus": 8,
        "onHit": [
            {
                "op": "damage",
                "amount": {
                    "dice": {
                        "count": 2,
                        "sides": 8
                    },
                    "bonus": 4
                },
                "damageType": "bludgeoning"
            },
            {
                "op": "grapple",
                "escapeDC": 16,
                "capacity": 2,
                "reachFeet": 10,
                "maximumSize": "medium",
                "dependentCondition": "incapacitated"
            }
        ]
    },
    "dependentCondition": "incapacitated"
}
},
{
    "kind": "action",
    "id": "devour-brain",
    "name": "Devour Brain",
    "target": {
        "side": "enemy",
        "rangeFeet": 5,
        "maximumSize": "medium",
        "requiresOwnedGrapple": true,
        "requiresCondition": "incapacitated",
        "excludeCreatureTypes": [
            "aberration",
            "beast",
            "celestial",

```

```

        "construct",
        "dragon",
        "elemental",
        "fey",
        "fiend",
        "giant",
        "monstrosity",
        "ooze",
        "plant",
        "undead"
    ]
},
"resolution": {
    "kind": "attack",
    "attackType": "melee",
    "attackBonus": 8,
    "onHit": {
        "op": "damage",
        "amount": {
            "dice": {
                "count": 6,
                "sides": 10
            },
            "bonus": 4
        },
        "damageType": "piercing",
        "onZeroHitPoints": "kill"
    }
},
{
    "kind": "action",
    "id": "confusing-display",
    "name": "Confusing Display",
    "target": {
        "side": "enemy",
        "rangeFeet": 30,
        "requiresSight": true,
        "requiresTargetSight": true,
        "excludesCondition": "confused"
    },
    "resolution": {
        "kind": "savingThrow",
        "ability": "wisdom",
        "dc": 16,
        "onFailure": {
            "op": "applyCondition",
            "condition": "confused",
            "duration": "repeatSaveEndOfTurn",
            "repeatSave": {
                "ability": "wisdom",
                "dc": 16
            },
            "maxTargetTurns": 10
        }
    },
    "activation": "bonusAction",
    "magical": true
}
]
}

```

### 3. Umbral Dragon

Download the complete Umbral Dragon package

**Start with:** `rending-assault`, `umbral-bite`, `umbral-claw`, `umbral-breath`, `umbral-line`, `thought-shearing-breath`.

**How it fits together:** Rending Assault combines a bite and claws in an ordered sequence. The bite has more than one damage type. The necrotic breath has cone and inherited line variants, while Thought-Shearing Breath is a separate psychic cone. Both breaths spend the same breath resource, restored by a turn-start recharge rule.

**What to look for:** Follow a breath from its target area to its saving throw and damage. Using either breath empties the shared pool until it recharges. The psychic stun has repeat saves and a turn cap. Changing the necrotic shape does not change the psychic cone.

**Try changing it:** Choose cone and line on otherwise identical placements, and inspect which creatures fall inside the footprint. Put an ally in front of the dragon to see the meaning of side any. Change the recharge threshold only in the shared recharge rule. CR 10 is an authored design label, not a verified balance result.

#### Complete package: Umbral Dragon

```
{
  "format": "encounterterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "1c4dc752-5fb0-4efb-a11f-f77233a1e76d",
    "name": "Umbral Dragon - Dream-Eater Breath Showcase",
    "version": "0.1.1"
  },
  "requires": { "ruleset": "dnd-2024", "srdCatalog": "5.2.1-encounterterra.1" },
  "definitions": [
    {
      "kind": "monster", "id": "umbral-dragon", "name": "Umbral Dragon (Breath Showcase)",
      "stats": {
        "challengeRating": "10", "hitPoints": 158, "armorClass": 17,
        "speedFeet": 40, "flySpeedFeet": 80, "darkvisionFeet": 120,
        "size": "large", "creatureType": "dragon",
        "abilityScores": { "strength": 21, "dexterity": 12, "constitution": 19, "intelligence": 13, "wisdom": 14,
"charisma": 17 },
        "savingThrows": { "strength": 5, "dexterity": 5, "constitution": 8, "intelligence": 1, "wisdom": 6,
"charisma": 7 },
        "difficultyClass": 16, "resistances": ["necrotic"]
      },
      "resources": [{ "id": "breath", "name": "Umbral Breath", "maximum": 1, "refresh": "longRest" }],
      "actions": ["#rending-assault", "#thought-shearing-breath"],
      "choices": [{
        "id": "breath-shape", "name": "Breath shape", "default": "cone",
        "options": [
          { "id": "cone", "name": "30-foot cone", "grants": ["#umbral-breath"] },
          { "id": "line", "name": "60-foot line", "grants": ["#umbral-line"] }
        ]
      }],
      "rules": [{
        "id": "recharge-breath", "event": "turn.start",
        "do": [{
          "op": "restoreResource", "resource": "breath",
          "amount": { "kind": "max", "operands": [0, {
```

```

        "kind": "subtract", "operands": [{ "kind": "dice", "dice": { "count": 1, "sides": 6 } }, 4]
      ] ] }
    ] ]
  },
  {
    "kind": "action", "id": "rending-assault", "name": "Rending Assault",
    "activation": { "economy": "action", "countsAs": "multiattack" },
    "target": { "side": "enemy", "rangeFeet": 10 },
    "resolution": { "kind": "sequence", "steps": [
      { "kind": "reference", "ref": "#umbral-bite" },
      { "repeat": 2, "resolve": { "kind": "reference", "ref": "#umbral-claw" } }
    ] }
  },
  {
    "kind": "action", "id": "umbral-bite", "name": "Umbral Bite",
    "resolution": { "kind": "attack", "attackBonus": 9, "damage": [
      { "dice": { "count": 2, "sides": 10 }, "bonus": 5, "damageType": "piercing" },
      { "dice": { "count": 1, "sides": 6 }, "damageType": "necrotic" }
    ] }
  },
  {
    "kind": "action", "id": "umbral-claw", "name": "Umbral Claw",
    "resolution": { "kind": "attack", "attackBonus": 9,
      "damage": { "dice": { "count": 2, "sides": 6 }, "bonus": 5, "damageType": "slashing" }
    }
  },
  {
    "kind": "action", "id": "umbral-breath", "name": "Umbral Breath (Cone)",
    "activation": "action", "resource": { "ref": "breath", "cost": 1 },
    "target": { "side": "any", "mode": "all", "area": { "shape": "cone", "sizeFeet": 30 } },
    "resolution": { "kind": "savingThrow", "ability": "dexterity", "dc": 16, "halfOnSave": true,
      "damage": { "dice": { "count": 8, "sides": 8 }, "damageType": "necrotic" }
    }
  },
  {
    "kind": "action", "id": "thought-shearing-breath", "name": "Thought-Shearing Breath",
    "activation": "action", "magical": true, "resource": { "ref": "breath", "cost": 1 },
    "target": { "side": "any", "mode": "all", "area": { "shape": "cone", "sizeFeet": 30 } },
    "resolution": { "kind": "savingThrow", "ability": "intelligence", "dc": 16,
      "onFailure": [
        { "op": "damage", "amount": { "dice": { "count": 5, "sides": 8 } }, "damageType": "psychic" },
        { "op": "applyCondition", "condition": "stunned", "duration": "repeatSaveEndOfTurn", "maxTargetTurns":
10,
          "repeatSave": { "ability": "intelligence", "dc": 16 } }
      ]
    }
  },
  {
    "kind": "action", "id": "umbral-line", "name": "Umbral Breath (Line)",
    "extends": "#umbral-breath",
    "replace": { "target": { "side": "any", "mode": "all", "area": { "shape": "line", "sizeFeet": 60,
"widthFeet": 5 } } }
  }
]
}

```

## 4. Blood Druid

Download the complete Blood Druid package

**Start with:** `blood-drain`, `thorn-whip`, `chill-touch`, `poison-spray`, `blight`, `withering-pollen`, `gloom-sprout`.

**How it fits together:** Blood Drain is a Bonus Action with a DC 16 Constitution save and 3d6 necrotic damage. One lifeDrain operation links damage taken, maximum-HP loss, and healing. Funeral Seed can create a local Gloom Sprout after that operation's eligible kill. The named spells and the original Withering Pollen are separate actions. Blighted Grove shares the pollen pool: one use creates a stationary concentration zone whose grove-withering callback resolves on entry and turn start, at most once per creature per turn.

**What to look for:** Follow the drain's damage and recovery together. Resistance changes both the maximum-HP loss and healing; a separate Blight kill cannot trigger Funeral Seed. Thorn Whip's size restriction belongs to the pull, so larger targets can still take damage. Chill Touch's healing prevention ends on the source's turn boundary.

**Try changing it:** Change Blood Drain to 2d6 without changing its other effects, then compare the result. Try a Construct and an Undead to see the exclusions, and fill the sprout cap before another eligible kill. The spells have explicit components, schools, and damage; they do not scale automatically or grant Wild Shape. CR 8 is a balance target.

### Complete package: Blood Druid

```
{
  "format": "encounterterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "59cde112-8464-4d73-a355-d5778609d20b",
    "name": "Blood Druid - Gardener of Blighted Groves",
    "version": "0.1.5"
  },
  "requires": { "ruleset": "dnd-2024", "srdCatalog": "5.2.1-encounterterra.1" },
  "definitions": [
    {
      "kind": "monster", "id": "blood-druid", "name": "Blood Druid",
      "stats": {
        "challengeRating": "8", "hitPoints": 119, "armorClass": 16,
        "speedFeet": 30, "size": "medium", "creatureType": "humanoid",
        "difficultyClass": 16, "spellAttackBonus": 8,
        "abilityScores": { "strength": 12, "dexterity": 14, "constitution": 16, "intelligence": 12, "wisdom": 20,
"charisma": 10 },
        "savingThrows": { "strength": 1, "dexterity": 2, "constitution": 6, "intelligence": 1, "wisdom": 8,
"charisma": 0 }
      },
      "resources": [
        { "id": "blight", "maximum": 2, "refresh": "longRest" },
        { "id": "pollen", "maximum": 2, "refresh": "longRest" }
      ],
      "actions": ["#blood-drain", "#thorn-whip", "#chill-touch", "#blight", "#poison-spray", "#withering-pollen",
"#blighted-grove"]
    },
    {
      "kind": "action", "id": "blood-drain", "name": "Blood Drain",
      "activation": "bonusAction", "magical": true,
      "target": { "side": "enemy", "rangeFeet": 30, "requiresSight": true, "excludeCreatureTypes": ["construct",
"undead"] }
    }
  ]
}
```

```

    "resolution": {
      "kind": "savingThrow", "ability": "constitution", "dc": 16,
      "onFailure": {
        "op": "lifeDrain", "amount": { "dice": { "count": 3, "sides": 6 } }, "damageType": "necrotic",
        "onKillSummon": { "ref": "#gloom-sprout", "creatureTypes": ["humanoid"], "maximumSize": "medium",
"maxActive": 1, "rangeFeet": 5 }
      }
    },
    {
      "kind": "spell", "id": "thorn-whip", "name": "Thorn Whip",
      "school": "transmutation",
      "components": ["verbal", "somatic", "material"],
      "level": 0, "activation": "action",
      "target": { "side": "enemy", "rangeFeet": 30 },
      "resolution": {
        "kind": "attack", "attackType": "melee", "attackBonus": 8,
        "onHit": [
          { "op": "damage", "amount": { "dice": { "count": 3, "sides": 6 } }, "damageType": "piercing" },
          { "op": "pull", "distanceFeet": 10, "maximumSize": "large" }
        ]
      }
    },
    {
      "kind": "spell", "id": "chill-touch", "name": "Chill Touch",
      "school": "necromancy",
      "components": ["verbal", "somatic"],
      "level": 0, "activation": "action",
      "target": { "side": "enemy", "rangeFeet": 5 },
      "resolution": {
        "kind": "attack", "attackType": "melee", "attackBonus": 8,
        "onHit": [
          { "op": "damage", "amount": { "dice": { "count": 3, "sides": 10 } }, "damageType": "necrotic" },
          { "op": "applyCondition", "condition": "cannot-regain-hit-points", "duration": "endOfSourceNextTurn" }
        ]
      }
    },
    {
      "kind": "spell", "id": "blight", "name": "Blight",
      "school": "necromancy",
      "components": ["verbal", "somatic"],
      "level": 4, "activation": "action", "resource": { "ref": "blight", "cost": 1 },
      "target": { "side": "enemy", "rangeFeet": 30, "requiresSight": true },
      "resolution": {
        "kind": "savingThrow", "ability": "constitution", "dc": 16, "halfOnSave": true,
        "autoFailCreatureTypes": ["plant"],
        "damage": { "dice": { "count": 8, "sides": 8 }, "damageType": "necrotic" }
      }
    },
    {
      "kind": "action", "id": "withering-pollen", "name": "Withering Pollen",
      "activation": "action", "magical": true, "resource": { "ref": "pollen", "cost": 1 },
      "target": { "side": "enemy", "rangeFeet": 60, "requiresSight": true },
      "resolution": {
        "kind": "savingThrow", "ability": "constitution", "dc": 16,
        "onFailure": [
          { "op": "damage", "amount": { "dice": { "count": 3, "sides": 8 } }, "damageType": "poison" },
          { "op": "applyCondition", "condition": "poisoned", "duration": "endOfSourceNextTurn" },
          { "op": "applyCondition", "condition": "cannot-regain-hit-points", "duration": "endOfSourceNextTurn" }
        ]
      }
    },
    {

```

```

    "kind": "monster", "id": "gloam-sprout", "name": "Gloam Sprout",
    "stats": {
      "challengeRating": "1/2", "hitPoints": 18, "armorClass": 12, "speedFeet": 20,
      "size": "small", "creatureType": "plant", "darkvisionFeet": 60,
      "resistances": ["necrotic"],
      "abilityScores": { "strength": 14, "dexterity": 10, "constitution": 12, "intelligence": 3, "wisdom": 10,
        "charisma": 3 }
    },
    "actions": ["#gloam-thorn"]
  },
  {
    "kind": "action", "id": "gloam-thorn", "name": "Gloam Thorn",
    "target": { "side": "enemy", "rangeFeet": 5 },
    "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 4,
      "damage": { "dice": { "count": 1, "sides": 6 }, "bonus": 2, "damageType": "piercing" }
    }
  },
  {
    "kind": "spell", "id": "poison-spray", "name": "Poison Spray",
    "school": "necromancy",
    "components": ["verbal", "somatic"],
    "level": 0, "activation": "action",
    "target": { "side": "enemy", "rangeFeet": 30 },
    "resolution": {
      "kind": "attack", "attackType": "ranged", "attackBonus": 8,
      "damage": { "dice": { "count": 3, "sides": 12 }, "damageType": "poison" }
    }
  },
  {
    "kind": "action", "id": "blighted-grove", "name": "Blighted Grove",
    "magical": true, "concentration": true, "target": { "side": "self" },
    "resource": { "ref": "pollen", "cost": 1 },
    "resolution": { "op": "createArea", "zone": {
      "area": { "shape": "radius", "sizeFeet": 15 }, "anchor": "point", "rangeFeet": 60,
      "durationRounds": 3, "oncePerTurn": true, "difficultTerrain": true,
      "affects": { "side": "enemy" }, "onEnter": "#grove-withering", "onTurnStart": "#grove-withering"
    } }
  },
  {
    "kind": "action", "id": "grove-withering", "name": "Grove Withering",
    "resolution": { "kind": "savingThrow", "ability": "constitution", "dc": 16,
      "halfOnSave": true, "onFailure": {
        "op": "damage", "amount": { "dice": { "count": 3, "sides": 6 } }, "damageType": "necrotic"
      } }
  }
]
}

```

## 5. Hushbell Colossus

Download the complete Hushbell Colossus package

**Start with:** `double-piston`, `afterbeat`, `unmaking-chime`, `final-toll`, `hushfield-radius`, `hushfield-cone`.

**How it fits together:** The physical piston attacks remain usable while the attached field suppresses magical chimes. Hushfield defaults to a radius, with a cone choice. Unmaking Chime and Final Toll attach different zero-HP consequences to their damage. Legendary Action points and Legendary Resistance uses are independent pools.

**What to look for:** The radius suppresses the colossus's own magic too. Suppression is not removal: a tracked magical condition can become active again outside the field. Unmaking Chime destroys remains only after its own lethal damage; Final Toll kills without that consequence.

**Try changing it:** Switch hushfield-shape to cone and compare the field's footprint. Adjust its sizeFeet to change its extent. Try reducing Legendary Resistance uses without changing Legendary Action points to see the separate budgets. The disintegration operation affects creatures and their remains, not arbitrary objects or equipment.

### Complete package: Hushbell Colossus

```
{
  "format": "encounterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "43fe1958-9e54-4b2b-a618-65b82c40fa57",
    "name": "Hushbell Colossus – Walking Sanctuary",
    "version": "0.1.0"
  },
  "requires": {
    "ruleset": "dnd-2024",
    "srdCatalog": "5.2.1-encounterra.1"
  },
  "definitions": [
    {
      "kind": "monster",
      "id": "hushbell-colossus",
      "name": "Hushbell Colossus",
      "stats": {
        "challengeRating": "12",
        "hitPoints": 187,
        "armorClass": 18,
        "speedFeet": 25,
        "size": "huge",
        "creatureType": "construct",
        "darkvisionFeet": 120,
        "difficultyClass": 17,
        "legendaryActionUses": 2,
        "legendaryResistanceUses": 2,
        "abilityScores": {
          "strength": 22,
          "dexterity": 9,
          "constitution": 20,
          "intelligence": 8,
          "wisdom": 15,
          "charisma": 12
        },
        "savingThrows": {
```

```

    "strength": 6,
    "dexterity": -1,
    "constitution": 9,
    "intelligence": -1,
    "wisdom": 6,
    "charisma": 1
  },
  "immunities": [
    "poison"
  ],
  "conditionImmunities": [
    "poisoned",
    "charmed",
    "frightened"
  ]
},
"resources": [
  {
    "id": "unmaking",
    "name": "Unmaking Chime",
    "maximum": 1,
    "refresh": "longRest"
  },
  {
    "id": "final-toll",
    "name": "Final Toll",
    "maximum": 2,
    "refresh": "longRest"
  }
],
"rules": [{
  "id": "recharge-unmaking", "event": "turn.start",
  "do": [{ "op": "restoreResource", "resource": "unmaking",
    "amount": { "kind": "max", "operands": [0, {
      "kind": "subtract", "operands": [{ "kind": "dice", "dice": { "count": 1, "sides": 6 } }, 4]
    } ] }
  }
}],
"choices": [
  {
    "id": "hushfield-shape",
    "name": "Hushfield shape",
    "options": [
      {
        "id": "radius",
        "grants": [
          "#hushfield-radius"
        ]
      },
      {
        "id": "cone",
        "grants": [
          "#hushfield-cone"
        ]
      }
    ],
    "default": "radius"
  }
],
"actions": [
  "#double-piston",
  "#unmaking-chime",
  "#final-toll",

```

```

      "#afterbeat"
    ]
  },
  {
    "kind": "action",
    "id": "piston-hammer",
    "name": "Piston Hammer",
    "target": {
      "side": "enemy",
      "rangeFeet": 10
    },
    "resolution": {
      "kind": "attack",
      "attackType": "melee",
      "attackBonus": 10,
      "onHit": {
        "op": "damage",
        "amount": {
          "dice": {
            "count": 2,
            "sides": 10
          },
          "bonus": 6
        },
        "damageType": "bludgeoning"
      }
    }
  },
  {
    "kind": "action",
    "id": "double-piston",
    "name": "Double Piston",
    "target": {
      "side": "enemy",
      "rangeFeet": 10
    },
    "resolution": {
      "kind": "sequence",
      "repeat": 2,
      "steps": [
        {
          "kind": "reference",
          "ref": "#piston-hammer"
        }
      ]
    },
    "activation": {
      "economy": "action",
      "countsAs": "multiattack"
    }
  },
  {
    "kind": "action",
    "id": "afterbeat",
    "name": "Afterbeat",
    "target": {
      "side": "enemy",
      "rangeFeet": 10
    },
    "resolution": {
      "kind": "reference",
      "ref": "#piston-hammer"
    }
  },

```

```

    "activation": "legendaryAction",
    "legendaryCost": 1
  },
  {
    "kind": "action",
    "id": "unmaking-chime",
    "name": "Unmaking Chime",
    "target": {
      "side": "enemy",
      "rangeFeet": 90,
      "requiresSight": true
    },
    "resolution": {
      "kind": "savingThrow",
      "ability": "dexterity",
      "dc": 17,
      "onFailure": {
        "op": "damage",
        "amount": {
          "dice": {
            "count": 7,
            "sides": 8
          }
        },
        "damageType": "force",
        "onZeroHitPoints": "disintegrate"
      }
    },
    "magical": true,
    "resource": {
      "ref": "unmaking",
      "cost": 1
    }
  },
  {
    "kind": "action",
    "id": "final-toll",
    "name": "Final Toll",
    "target": {
      "side": "enemy",
      "rangeFeet": 60,
      "requiresSight": true
    },
    "resolution": {
      "kind": "savingThrow",
      "ability": "constitution",
      "dc": 17,
      "onFailure": {
        "op": "damage",
        "amount": {
          "dice": {
            "count": 6,
            "sides": 10
          }
        },
        "damageType": "necrotic",
        "onZeroHitPoints": "kill"
      }
    },
    "magical": true,
    "resource": {
      "ref": "final-toll",
      "cost": 1
    }
  }

```

```

    }
  },
  {
    "kind": "action",
    "id": "hushfield-radius",
    "name": "Hushfield (radius)",
    "target": {
      "side": "enemy",
      "rangeFeet": 90
    },
    "resolution": {
      "op": "antimagicArea",
      "target": "owner",
      "area": {
        "shape": "radius",
        "sizeFeet": 30
      },
      "duration": "startOfSourceNextTurn"
    },
    "activation": "bonusAction"
  },
  {
    "kind": "action",
    "id": "hushfield-cone",
    "name": "Hushfield (cone)",
    "target": {
      "side": "enemy",
      "rangeFeet": 90
    },
    "resolution": {
      "op": "antimagicArea",
      "target": "owner",
      "area": {
        "shape": "cone",
        "sizeFeet": 30
      },
      "duration": "startOfSourceNextTurn"
    },
    "activation": "bonusAction"
  }
]
}

```

## 6. Marbleweaver Moth

Download the complete Marbleweaver Moth package

**Start with:** `binding-filament`, `leaden-dust`, `calcifying-cocoon`, `mineral-mandibles`.

**How it fits together:** Binding Filament applies repeat-save restraint. Leaden Dust uses the narrower dragon-slowed condition. Calcifying Cocoon starts with restraint, then changes that condition to Petrified on the first failed repeat save. These are different control patterns, not interchangeable labels.

**What to look for:** A successful repeat save ends the cocoon's initial stage. Removing that stage before the next failure prevents petrification. The final condition keeps its magical origin, has no further saves, and lasts until removed or the encounter ends. It is not permanent campaign state.

**Try changing it:** Compare dragon-slowed with slowed before choosing the effect you want. The narrow version does not include the full Slow spell's AC, save, attack-count, or casting penalties. Change the cocoon DC and observe the transition. Ordinary flight does not imply hover, and binding does not lift objects.

### Complete package: Marbleweaver Moth

```
{
  "format": "encounterterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "cb80fe29-6dfa-439d-9d26-fb341afc351e",
    "name": "Marbleweaver Moth - Quarry Trapper",
    "version": "0.1.0"
  },
  "requires": { "ruleset": "dnd-2024", "srdCatalog": "5.2.1-encounterterra.1" },
  "definitions": [
    {
      "kind": "monster", "id": "marbleweaver-moth", "name": "Marbleweaver Moth",
      "stats": {
        "hitPoints": 112, "armorClass": 15, "size": "large", "creatureType": "monstrosity",
        "speedFeet": 15, "flySpeedFeet": 40, "darkvisionFeet": 60, "difficultyClass": 15,
        "abilityScores": { "strength": 16, "dexterity": 16, "constitution": 18, "intelligence": 5, "wisdom": 14,
"charisma": 8 },
        "savingThrows": { "strength": 3, "dexterity": 6, "constitution": 7, "intelligence": -3, "wisdom": 2,
"charisma": -1 }
      },
      "resources": [{ "id": "cocoon", "name": "Calcifying Cocoon", "maximum": 2, "refresh": "longRest" }],
      "actions": ["#mineral-mandibles", "#binding-filament", "#leaden-dust", "#calcifying-cocoon"]
    },
    {
      "kind": "action", "id": "mineral-mandibles", "name": "Mineral Mandibles",
      "target": { "side": "enemy", "rangeFeet": 5 },
      "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 6,
        "damage": { "dice": { "count": 3, "sides": 6 }, "bonus": 3, "damageType": "piercing" } }
    },
    {
      "kind": "action", "id": "binding-filament", "name": "Binding Filament", "magical": true,
      "activation": "bonusAction",
      "target": { "side": "enemy", "rangeFeet": 60, "requiresSight": true, "excludesCondition": "restrained" },
      "resolution": { "kind": "savingThrow", "ability": "strength", "dc": 15,
        "onFailure": { "op": "applyCondition", "condition": "restrained", "duration": "repeatSaveEndOfTurn",
          "repeatSave": { "ability": "strength", "dc": 15 } } }
    },
  ],
}
```

```

{
  "kind": "action", "id": "leaden-dust", "name": "Leaden Dust", "magical": true,
  "target": { "side": "enemy", "rangeFeet": 60, "requiresSight": true, "excludesCondition": "dragon-slowed" },
  "resolution": { "kind": "savingThrow", "ability": "constitution", "dc": 15, "halfOnSave": true,
    "onFailure": [
      { "op": "damage", "amount": { "dice": { "count": 4, "sides": 8 } }, "damageType": "necrotic" },
      { "op": "applyCondition", "condition": "dragon-slowed", "duration": "endOfTargetNextTurn" }
    ],
    "onSuccess": { "op": "damage", "amount": { "kind": "divide", "operands": [{ "dice": { "count": 4, "sides":
8 } }, 2] }, "damageType": "necrotic" }
  },
  {
    "kind": "action", "id": "calcifying-cocoon", "name": "Calcifying Cocoon", "magical": true,
    "resource": { "ref": "cocoon", "cost": 1 },
    "target": { "side": "enemy", "rangeFeet": 60, "requiresSight": true, "excludesCondition": "petrified" },
    "resolution": { "kind": "savingThrow", "ability": "constitution", "dc": 15,
      "onFailure": { "op": "applyCondition", "condition": "restrained", "duration": "repeatSaveEndOfTurn",
        "repeatSave": { "ability": "constitution", "dc": 15, "onFailureCondition": "petrified" } }
    }
  }
]
}

```

## 7. Dawnward Sentinel

Download the complete Dawnward Sentinel package

**Start with:** `armored-sentinel`, `sentinel-training`, `decisive-blows`, `first-light`, `sunedge`, `dawnward-ward`, `rekindle`.

**How it fits together:** Equipment declares heavy armor and a shield, already accounted for in base AC 18. Defensive Training adds another point while armor is worn. Sentinel Training forwards the accuracy and power parameters to Sunedge. The weapon has a 19–20 critical range. Dawnward Ward and Rekindle draw on the same limited resolve pool.

**What to look for:** Separate the permanent AC grant from the automatic defensive Reaction. Ward costs two resolve and cannot be used with only one remaining. Sunedge can also make opportunity attacks, which compete for the Reaction. A critical doubles its dice count, not the flat power argument.

**Try changing it:** Change a training argument and inspect Sunedge's numbers. Choose Decisive Blows to add the first-hit weapon-dice selection: compare two whole 2d8 totals, or two 4d8 totals on a critical. Set `criticalThreshold` to 20 to remove only the expanded range. Level 8 by itself gives no Fighter features.

### Complete package: Dawnward Sentinel

```
{
  "format": "encounterterra.homebrew",
  "schemaVersion": "0.1",
  "package": { "id": "eaf098a5-089d-40bc-9dbf-bb1a7a4f8d97", "name": "Dawnward Sentinel – Protector of the Last Light", "version": "0.1.2" },
  "requires": { "ruleset": "dnd-2024", "srdCatalog": "5.2.1-encounterterra.1" },
  "definitions": [
    {
      "kind": "hero", "id": "dawnward-sentinel", "name": "Dawnward Sentinel",
      "equipment": { "armor": "heavy", "shield": true },
      "stats": {
        "level": 8, "hitPoints": 80, "armorClass": 18, "speedFeet": 30, "size": "medium", "creatureType": "humanoid",
        "abilityScores": { "strength": 20, "dexterity": 10, "constitution": 16, "intelligence": 10, "wisdom": 14, "charisma": 14 },
        "savingThrows": { "strength": 8, "dexterity": 0, "constitution": 6, "intelligence": 0, "wisdom": 2, "charisma": 2 },
        "difficultyClass": 15
      },
      "resources": [{ "id": "resolve", "maximum": 6, "refresh": "shortRest" }],
      "grants": [{ "ref": "#sentinel-training", "args": { "accuracy": 8, "power": 5 } }, "#first-light", "#armored-sentinel"],
      "actions": ["#dawnward-ward", "#rekindle"],
      "choices": [{
        "id": "weapon-training", "name": "Weapon Training", "default": "standard",
        "options": [
          { "id": "standard", "name": "Standard", "grants": [] },
          { "id": "decisive-blows", "name": "Decisive Blows", "grants": ["#decisive-blows"] }
        ]
      }]
    },
    {
      "kind": "feat", "id": "decisive-blows", "name": "Decisive Blows",
      "modifiers": [{ "kind": "weaponDamageRollTwice" }]
    }
  ]
}
```

```

    "kind": "feat", "id": "armored-sentinel", "name": "Defensive Training",
    "modifiers": [{ "kind": "armorClass", "amount": 1, "whenEquipment": { "armor": ["light", "medium", "heavy"] }
  } ]
},
{
  "kind": "ability", "id": "sentinel-training", "name": "Sentinel Training",
  "parameters": {
    "accuracy": { "valueType": "integer", "minimum": 0, "maximum": 20 },
    "power": { "valueType": "integer", "minimum": 0, "maximum": 20 }
  },
  "grants": [{ "ref": "#sunedge", "args": { "accuracy": { "param": "accuracy" }, "power": { "param": "power" } }
  } ]
},
{
  "kind": "feat", "id": "first-light", "name": "First Light",
  "rules": [{ "id": "ready-the-shield", "event": "encounter.start", "oncePer": "encounter",
    "do": [{ "op": "temporaryHitPoints", "target": "self", "amount": 3 } ]
  } ]
},
{
  "kind": "action", "id": "sunedge", "name": "Sunedge", "activation": "action",
  "delivery": "weapon", "weapon": { "opportunityAttack": true, "type": "melee", "category": "martial",
  "properties": ["versatile"], "twoHanded": false },
  "parameters": {
    "accuracy": { "valueType": "integer", "minimum": 0, "maximum": 20 },
    "power": { "valueType": "integer", "minimum": 0, "maximum": 20 }
  },
  "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
  "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": { "param": "accuracy" },
  "criticalThreshold": 19,
  "onHit": { "op": "damage", "amount": { "kind": "add", "operands": [{ "dice": { "count": 2, "sides": 8 } },
  { "param": "power" } ] }, "damageType": "slashing" }
  }
},
{
  "kind": "action", "id": "dawnward-ward", "name": "Dawnward Ward", "activation": "reaction", "trigger":
  "damage.before",
  "resource": { "ref": "resolve", "cost": 2 }, "target": { "side": "self" },
  "resolution": { "op": "temporaryHitPoints", "amount": { "dice": { "count": 1, "sides": 8 }, "bonus": 4 } }
},
{
  "kind": "action", "id": "rekindle", "name": "Rekindle", "activation": "bonusAction",
  "resource": { "ref": "resolve", "cost": 2 }, "target": { "side": "self" },
  "resolution": { "op": "heal", "amount": { "dice": { "count": 2, "sides": 8 }, "bonus": 4 } }
}
]
}

```

## 8. Emberbound Adept

Download the complete Emberbound Adept package

**Start with:** `fire-training`, `cold-training`, `fire-lance`, `cold-lance`, `cinder-burst`, `rime-burst`.

**How it fits together:** The element choice gives an entire training bundle. That bundle forwards its accuracy parameter to the selected spells. Cold definitions inherit and replace the fire spells' damage payloads rather than just changing their names. Payment alternatives let the same burst use its daily charge or an available spell slot.

**What to look for:** The cold choice gives cold actions, not both elemental kits. Read each payment alongside its owning resource. The burst's explicit formula rolls 6d6 at third level or 7d6 at fourth level; a daily cast uses the spell's printed third level.

**Try changing it:** Change the default element, or choose another element when placing the hero. Compare the spells against fire-resistant and cold-resistant targets. Exhaust the daily charge and see the remaining slot alternatives. The written slots refresh on a long rest; the level label does not add Wizard features or prepared spells.

### Complete package: Emberbound Adept

```
{
  "format": "encounterra.homebrew",
  "schemaVersion": "0.1",
  "package": {
    "id": "d70ba198-7fca-4a83-bbca-8342f2050c80",
    "name": "Emberbound Adept – The Twofold Flame",
    "version": "0.1.3"
  },
  "requires": {
    "ruleset": "dnd-2024",
    "srdCatalog": "5.2.1-encounterra.1"
  },
  "definitions": [
    {
      "kind": "hero",
      "id": "emberbound-adept",
      "name": "Emberbound Adept",
      "stats": {
        "level": 8,
        "hitPoints": 54,
        "armorClass": 15,
        "speedFeet": 30,
        "size": "medium",
        "creatureType": "humanoid",
        "abilityScores": {
          "strength": 10,
          "dexterity": 14,
          "constitution": 14,
          "intelligence": 18,
          "wisdom": 12,
          "charisma": 10
        }
      },
      "savingThrows": {
        "strength": 0,
        "dexterity": 2,
        "constitution": 2,
        "intelligence": 7,
```

```

        "wisdom": 4,
        "charisma": 0
    },
    "difficultyClass": 15,
    "spellAttackBonus": 7
},
"resources": [
    {
        "id": "daily-burst",
        "maximum": 1,
        "refresh": "longRest"
    },
    {
        "id": "spell-slots",
        "kind": "spellSlots",
        "slots": { "3": 2, "4": 1 },
        "refresh": "longRest"
    }
],
"choices": [
    {
        "id": "element",
        "name": "Bound element",
        "default": "fire",
        "options": [
            {
                "id": "fire",
                "grants": [
                    {
                        "ref": "#fire-training",
                        "args": {
                            "accuracy": 7
                        }
                    }
                ]
            },
            {
                "id": "cold",
                "grants": [
                    {
                        "ref": "#cold-training",
                        "args": {
                            "accuracy": 7
                        }
                    }
                ]
            }
        ]
    }
],
{
    "kind": "ability",
    "id": "fire-training",
    "name": "Flame Training",
    "parameters": {
        "accuracy": {
            "valueType": "integer",
            "minimum": 0,
            "maximum": 20
        }
    }
},
"grants": [

```

```

    {
      "ref": "#fire-lance",
      "args": {
        "accuracy": {
          "param": "accuracy"
        }
      }
    },
    "#cinder-burst"
  ]
},
{
  "kind": "ability",
  "id": "cold-training",
  "name": "Frost Training",
  "parameters": {
    "accuracy": {
      "valueType": "integer",
      "minimum": 0,
      "maximum": 20
    }
  },
  "grants": [
    {
      "ref": "#cold-lance",
      "args": {
        "accuracy": {
          "param": "accuracy"
        }
      }
    }
  ],
  "#rime-burst"
]
},
{
  "kind": "spell",
  "id": "fire-lance",
  "school": "evocation",
  "name": "Fire Lance",
  "level": 0,
  "activation": "action",
  "parameters": {
    "accuracy": {
      "valueType": "integer",
      "minimum": 0,
      "maximum": 20
    }
  },
  "target": {
    "side": "enemy",
    "rangeFeet": 120
  },
  "resolution": {
    "kind": "attack",
    "attackType": "ranged",
    "attackBonus": {
      "param": "accuracy"
    }
  },
  "onHit": {
    "op": "damage",
    "amount": {
      "dice": {
        "count": 2,

```

```

        "sides": 10
      }
    },
    "damageType": "fire"
  }
},
{
  "kind": "spell",
  "id": "cold-lance",
  "name": "Cold Lance",
  "extends": "#fire-lance",
  "replace": {
    "resolution": {
      "kind": "attack",
      "attackType": "ranged",
      "attackBonus": {
        "param": "accuracy"
      },
      "onHit": {
        "op": "damage",
        "amount": {
          "dice": {
            "count": 2,
            "sides": 10
          }
        },
        "damageType": "cold"
      }
    }
  }
},
{
  "kind": "spell",
  "id": "cinder-burst",
  "school": "evocation",
  "name": "Cinder Burst",
  "level": 3,
  "activation": "action",
  "target": {
    "side": "any",
    "mode": "all",
    "area": {
      "shape": "radius",
      "sizeFeet": 15
    }
  },
  "resolution": {
    "kind": "savingThrow",
    "ability": "dexterity",
    "dc": 15,
    "halfOnSave": true,
    "damage": {
      "dice": {
        "count": { "kind": "add", "operands": [3, { "kind": "castLevel" }] },
        "sides": 6
      },
      "damageType": "fire"
    }
  },
  "payments": [
    {
      "id": "daily",

```

```

    "mode": "daily",
    "resource": "daily-burst",
    "cost": 1
  },
  {
    "id": "third-level-slot",
    "mode": "spellSlot",
    "resource": "spell-slots",
    "slotLevel": 3
  },
  {
    "id": "fourth-level-slot",
    "mode": "spellSlot",
    "resource": "spell-slots",
    "slotLevel": 4
  }
]
},
{
  "kind": "spell",
  "id": "rime-burst",
  "name": "Rime Burst",
  "extends": "#cinder-burst",
  "replace": {
    "resolution": {
      "kind": "savingThrow",
      "ability": "dexterity",
      "dc": 15,
      "halfOnSave": true,
      "damage": {
        "dice": {
          "count": { "kind": "add", "operands": [3, { "kind": "castLevel" }] },
          "sides": 6
        },
        "damageType": "cold"
      }
    }
  }
}
}
]
}

```

## 9. Cinderroot Warden

Download the complete Cinderroot Warden package

**Start with:** `regrowth`, `heartwood-resolve`, `root-staff`, `sap-mending`, `heartseed-ward`.

**How it fits together:** Regrowth is a reusable rule with an amount and a blocked-by damage-type list. It queries received damage since the previous turn started; fire inhibits healing. Heartwood Resolve combines condition checks and a limited trigger. Sap and heartseed use separate short-rest and long-rest resources. Briar Shell shares heartseed with Heartseed Ward: a larger body, two weapon attacks and 16 temporary HP without replacing real HP. Original Regrowth remains active. Shed Shell, duration expiry or loss of temporary HP reverts the form.

**What to look for:** Receiving fire blocks regeneration, but dealing fire does not. The positive-HP check prevents this trait from reviving a downed owner. Healing and temporary HP behave differently: one restores current HP, while the other supplies a separate, non-stacking protective pool.

**Try changing it:** Change blocked-by to include radiant, or lower the regeneration amount. Compare matching and nonmatching incoming damage. The written Hit Dice pool is eight d10s; changing that pool does not recalculate the authored 84 HP. Try linked encounters with short and long rests to compare resource recovery.

### Complete package: Cinderroot Warden

```
{
  "format": "encounterra.homebrew",
  "schemaVersion": "0.1",
  "requires": {
    "ruleset": "dnd-2024",
    "srdCatalog": "5.2.1-encounterra.1"
  },
  "package": {
    "id": "272dcd06-a466-4c1c-ad41-f8ba3b7ead69",
    "name": "Cinderroot Warden – Living-Wood Guardian",
    "version": "0.1.2"
  },
  "definitions": [
    {
      "kind": "hero",
      "id": "cinderroot-warden",
      "name": "Cinderroot Warden",
      "stats": {
        "level": 8,
        "hitPoints": 84,
        "hitDice": { "count": 8, "size": 10 },
        "armorClass": 17,
        "speedFeet": 30,
        "burrowSpeedFeet": 10,
        "size": "medium",
        "creatureType": "humanoid",
        "abilityScores": {
          "strength": 18,
          "dexterity": 10,
          "constitution": 18,
          "intelligence": 10,
          "wisdom": 16,
          "charisma": 8
        }
      },
    },
  ],
}
```

```

    "savingThrows": {
      "strength": 7,
      "dexterity": 0,
      "constitution": 7,
      "intelligence": 0,
      "wisdom": 3,
      "charisma": -1
    },
    "difficultyClass": 15
  },
  "resources": [
    {
      "id": "sap",
      "maximum": 2,
      "refresh": "shortRest"
    },
    {
      "id": "heartseed",
      "maximum": 1,
      "refresh": "longRest"
    }
  ],
  "actions": [
    "#root-staff",
    "#sap-mending",
    "#heartseed-ward",
    "#briar-shell-form"
  ],
  "grants": [
    {
      "ref": "#regrowth",
      "args": {
        "amount": 6,
        "blocked-by": [
          "fire"
        ]
      }
    },
    "#heartwood-resolve"
  ]
},
{
  "kind": "ability",
  "id": "regrowth",
  "name": "Cinderroot Regrowth",
  "parameters": {
    "amount": {
      "valueType": "integer",
      "minimum": 0,
      "maximum": 20
    },
    "blocked-by": {
      "valueType": "damageTypeList"
    }
  },
  "rules": [
    {
      "id": "living-bark",
      "event": "turn.start",
      "oncePer": "turn",
      "when": {
        "all": [
          {

```

```

        "hitPoints": {
            "subject": "self",
            "operator": "above",
            "value": 0
        }
    },
    {
        "not": {
            "eventOccurred": {
                "event": "damage.received",
                "subject": "self",
                "since": "self.previousTurnStarted",
                "where": {
                    "damageType": {
                        "inParam": "blocked-by"
                    }
                }
            }
        }
    }
],
},
"do": [
    {
        "op": "heal",
        "target": "self",
        "amount": {
            "param": "amount"
        }
    }
]
}
],
},
{
    "kind": "effect",
    "id": "heartwood-resolve",
    "name": "Heartwood Resolve",
    "rule": {
        "event": "condition.applied",
        "limit": {
            "uses": 2,
            "scope": "encounter"
        },
    },
    "when": {
        "any": [
            {
                "hasCondition": "poisoned"
            },
            {
                "hasCondition": "frightened"
            }
        ]
    },
},
"do": [
    {
        "op": "removeCondition",
        "target": "self",
        "condition": "poisoned"
    },
    {
        "op": "removeCondition",
        "target": "self",

```

```

        "condition": "frightened"
      }
    ]
  }
},
{
  "kind": "action",
  "id": "root-staff",
  "name": "Root Staff",
  "activation": "action",
  "target": {
    "side": "enemy",
    "rangeFeet": 10
  },
  "resolution": {
    "kind": "attack",
    "attackType": "melee",
    "attackBonus": 7,
    "onHit": [
      {
        "op": "damage",
        "amount": {
          "dice": {
            "count": 2,
            "sides": 8
          },
          "bonus": 4
        },
        "damageType": "bludgeoning"
      },
      {
        "op": "push",
        "distanceFeet": 5,
        "maximumSize": "large"
      }
    ]
  }
},
{
  "kind": "action",
  "id": "sap-mending",
  "name": "Sap Mending",
  "activation": "bonusAction",
  "target": {
    "side": "self"
  },
  "resource": {
    "ref": "sap",
    "cost": 1
  },
  "resolution": {
    "op": "heal",
    "amount": {
      "kind": "add",
      "operands": [
        {
          "dice": {
            "count": 2,
            "sides": 6
          }
        },
        {
          "kind": "divide",

```

```

        "operands": [
            {
                "kind": "multiply",
                "operands": [
                    2,
                    8
                ]
            },
            2
        ]
    }
]
}
}
},
{
    "kind": "action",
    "id": "heartseed-ward",
    "name": "Heartseed Ward",
    "activation": "bonusAction",
    "target": {
        "side": "self"
    },
    "resource": {
        "ref": "heartseed",
        "cost": 1
    },
    "resolution": {
        "op": "temporaryHitPoints",
        "amount": {
            "kind": "min",
            "operands": [
                20,
                {
                    "kind": "max",
                    "operands": [
                        8,
                        {
                            "dice": {
                                "count": 3,
                                "sides": 6
                            }
                        }
                    ]
                }
            ]
        }
    }
}
},
{
    "kind": "action", "id": "briar-shell-form", "name": "Briar Shell",
    "activation": "bonusAction", "magical": true,
    "target": { "side": "self" }, "resource": { "ref": "heartseed", "cost": 1 },
    "resolution": { "op": "transform", "ref": "#briar-shell-body", "durationRounds": 10,
        "temporaryHitPoints": 16, "retainMentalAbilities": true,
        "endWhenTemporaryHitPointsLost": true, "clearTemporaryHitPointsOnEnd": true }
},
{
    "kind": "monster", "id": "briar-shell-body", "name": "Briar Shell Body",
    "stats": { "hitPoints": 60, "armorClass": 18, "size": "large", "speedFeet": 30,
        "burrowSpeedFeet": 15, "abilityScores": { "strength": 20, "dexterity": 10, "constitution": 18 },
        "savingThrows": { "strength": 8, "dexterity": 0, "constitution": 7 }, "attacksPerAction": 2 },
    "actions": ["#briar-slam", "#shed-shell"]
}

```

```

    },
    {
      "kind": "action", "id": "briar-slam", "name": "Briar Slam", "delivery": "weapon",
      "weapon": { "category": "natural", "attackAbility": "strength", "opportunityAttack": true },
      "target": { "side": "enemy", "rangeFeet": 5, "includeDowned": true },
      "resolution": { "kind": "attack", "attackType": "melee", "attackBonus": 8,
        "damage": { "dice": { "count": 2, "sides": 8 }, "bonus": 5, "damageType": "bludgeoning" } }
    },
    {
      "kind": "action", "id": "shed-shell", "name": "Shed Shell", "activation": "freeAction",
      "target": { "side": "self" }, "resolution": { "op": "revertForm" }
    }
  ]
}

```

## 10. Lantern Medium

Download the complete Lantern Medium package

**Start with:** beacon-bolt, kindly-mending, sheltering-light, clear-mind, lantern-step, call-lantern-spirit, lantern-spirit, guiding-chorus, chorus-effect.

**How it fits together:** The Medium combines an enemy attack with ally healing, temporary HP, a condition-gated cure, and self teleportation. Its guidance choice gives either Call Lantern Spirit or Guiding Chorus. The spirit is a complete local monster with its own action; the chorus attaches attack and save bonuses to up to three allies.

**What to look for:** Follow concentration through both alternatives. Losing concentration removes the spirit or the chorus attachments belonging to that cast. Healing can help an eligible downed ally, but temporary HP alone cannot revive one. Clear Mind is available only for a frightened target.

**Try changing it:** Place injured, downed, frightened, and healthy allies nearby. Compare calling with chorus and inspect the different support options. Chorus spends one oil for the entire group, not one per ally. Change maxTargets to two to reduce the group size. Teleportation needs a legal destination within its range and is not planar travel.

### Complete package: Lantern Medium

```
{
  "format": "encounterra.homebrew",
  "schemaVersion": "0.1",
  "requires": {
    "ruleset": "dnd-2024",
    "srdCatalog": "5.2.1-encounterra.1"
  },
  "package": {
    "id": "4c599268-e4ce-41cb-84b1-6884db9e0e4a",
    "name": "Lantern Medium – Keeper of Kindly Spirits",
    "version": "0.1.3"
  },
  "definitions": [
    {
      "kind": "hero",
      "id": "lantern-medium",
      "name": "Lantern Medium",
      "stats": {
        "level": 8,
        "hitPoints": 58,
        "armorClass": 15,
        "speedFeet": 30,
        "darkvisionFeet": 60,
        "size": "medium",
        "creatureType": "humanoid",
        "abilityScores": {
          "strength": 8,
          "dexterity": 14,
          "constitution": 14,
          "intelligence": 12,
          "wisdom": 18,
          "charisma": 14
        }
      },
      "savingThrows": {
        "strength": -1,
        "dexterity": 2,
```

```

        "constitution": 2,
        "intelligence": 1,
        "wisdom": 7,
        "charisma": 5
    },
    "difficultyClass": 15,
    "spellAttackBonus": 7
},
"resources": [
    {
        "id": "lantern-oil",
        "maximum": 4,
        "refresh": "shortRest"
    },
    {
        "id": "spirit-call",
        "maximum": 1,
        "refresh": "longRest"
    }
],
"actions": [
    "#beacon-bolt",
    "#kindly-mending",
    "#sheltering-light",
    "#clear-mind",
    "#lantern-step"
],
"choices": [{
    "id": "guidance", "name": "Spirit Guidance", "default": "calling",
    "options": [
        { "id": "calling", "name": "Call a Lantern Spirit", "grants": ["#call-lantern-spirit"] },
        { "id": "chorus", "name": "Guiding Chorus", "grants": ["#guiding-chorus"] }
    ]
}]
},
{
    "kind": "spell", "id": "guiding-chorus", "name": "Guiding Chorus",
    "level": 1, "school": "enchantment", "components": ["verbal"], "concentration": true,
    "target": { "side": "ally", "rangeFeet": 30, "includeSelf": true, "maxTargets": 3 },
    "resource": { "ref": "lantern-oil", "cost": 1 },
    "resolution": { "op": "applyEffect", "ref": "#chorus-effect" }
},
{
    "kind": "effect", "id": "chorus-effect", "name": "Kindly Guidance",
    "modifiers": [
        { "kind": "attackBonusDie", "dice": { "count": 1, "sides": 4 } },
        { "kind": "savingThrowBonusDie", "dice": { "count": 1, "sides": 4 } }
    ],
    "attachments": [{ "kind": "combatant", "duration": "concentration", "maxTargetTurns": 10 }]
},
{
    "kind": "spell",
    "id": "beacon-bolt",
    "school": "evocation",
    "name": "Beacon Bolt",
    "level": 0,
    "activation": "action",
    "target": {
        "side": "enemy",
        "rangeFeet": 90
    },
    "resolution": {
        "kind": "attack",

```

```

    "attackType": "ranged",
    "attackBonus": 7,
    "onHit": {
      "op": "damage",
      "amount": {
        "dice": {
          "count": 2,
          "sides": 8
        }
      },
      "damageType": "radiant"
    }
  },
  {
    "kind": "spell",
    "id": "kindly-mending",
    "school": "abjuration",
    "name": "Kindly Mending",
    "level": 1,
    "activation": "action",
    "target": {
      "side": "ally",
      "rangeFeet": 60,
      "includeDowned": true,
      "includeSelf": true
    },
    "resolution": {
      "op": "heal",
      "amount": {
        "dice": {
          "count": 2,
          "sides": 8
        }
      },
      "bonus": 4
    }
  },
  {
    "resource": {
      "ref": "lantern-oil",
      "cost": 1
    }
  },
  {
    "kind": "spell",
    "id": "sheltering-light",
    "school": "abjuration",
    "name": "Sheltering Light",
    "level": 1,
    "activation": "bonusAction",
    "target": {
      "side": "ally",
      "rangeFeet": 30
    },
    "resolution": {
      "op": "temporaryHitPoints",
      "amount": {
        "dice": {
          "count": 2,
          "sides": 6
        }
      },
      "bonus": 4
    }
  },
  {

```

```

    "resource": {
      "ref": "lantern-oil",
      "cost": 1
    }
  },
  {
    "kind": "spell",
    "id": "clear-mind",
    "school": "abjuration",
    "name": "Clear Mind",
    "level": 2,
    "activation": "action",
    "target": {
      "side": "ally",
      "rangeFeet": 30,
      "requiresCondition": "frightened"
    },
    "resolution": {
      "op": "removeCondition",
      "condition": "frightened"
    },
    "resource": {
      "ref": "lantern-oil",
      "cost": 1
    }
  },
  {
    "kind": "spell",
    "id": "lantern-step",
    "school": "conjuration",
    "name": "Lantern Step",
    "level": 2,
    "activation": "bonusAction",
    "target": {
      "side": "self"
    },
    "resolution": {
      "op": "teleport",
      "distanceFeet": 30,
      "requiresSight": true
    },
    "resource": {
      "ref": "lantern-oil",
      "cost": 1
    }
  },
  {
    "kind": "spell",
    "id": "call-lantern-spirit",
    "school": "conjuration",
    "name": "Call Lantern Spirit",
    "level": 3,
    "activation": "action",
    "target": {
      "side": "self"
    },
    "resolution": {
      "op": "summon",
      "ref": "#lantern-spirit",
      "count": 1,
      "rangeFeet": 10,
      "maxActive": 1
    },
  },

```

```

    "concentration": true,
    "resource": {
      "ref": "spirit-call",
      "cost": 1
    }
  },
  {
    "kind": "monster",
    "id": "lantern-spirit",
    "name": "Lantern Spirit",
    "stats": {
      "challengeRating": "1",
      "hitPoints": 26,
      "armorClass": 14,
      "speedFeet": 20,
      "flySpeedFeet": 30,
      "size": "small",
      "creatureType": "celestial",
      "darkvisionFeet": 60,
      "abilityScores": {
        "strength": 8,
        "dexterity": 16,
        "constitution": 12,
        "intelligence": 8,
        "wisdom": 14,
        "charisma": 12
      },
      "resistances": [
        "radiant"
      ]
    },
    "actions": [
      "#spirit-spark"
    ]
  },
  {
    "kind": "action",
    "id": "spirit-spark",
    "name": "Spirit Spark",
    "activation": "action",
    "magical": true,
    "target": {
      "side": "enemy",
      "rangeFeet": 30
    },
    "resolution": {
      "kind": "attack",
      "attackType": "ranged",
      "attackBonus": 5,
      "onHit": {
        "op": "damage",
        "amount": {
          "dice": {
            "count": 2,
            "sides": 6
          },
          "bonus": 3
        }
      },
      "damageType": "radiant"
    }
  }
}

```

```
]
}
```

Homebrew Beta v0.1 · Schema "0.1" · Guide updated 8 September 2026