BookStack D&D 5e Statblock Guide

This guide documents how to use the BookStack D&D 5e statblock renderer with inline JSON and Tetra-cube `.monster` files.

Install the renderer once in BookStack custom HTML head content. After that, each page can include one or more hidden JSON wrappers. When the page loads, the renderer replaces those wrappers with styled D&D 5e statblocks.

Files

- `bookstack-custom-html-with-json-statblocks.html`
  The main BookStack custom HTML/CSS/JS bundle.
- `bookstack-monster-paste-helper.html`
  A local helper page for loading, dropping, or pasting a `.monster` file and generating BookStack-ready wrapper HTML.
- `bookstack-dnd-statblock-guide.html`
  The full HTML guide with rendered previews.
- `bookstack-dnd-statblock-guide.md`
  This Markdown version.

Quick Start

1. Open `bookstack-custom-html-with-json-statblocks.html`.
2. Copy the statblock CSS and JavaScript into BookStack's **Custom HTML Head Content**.
3. Open a BookStack page in source/HTML mode.
4. Paste one of the wrapper snippets below.
5. Put the JSON from a `.monster` file inside the wrapper.
6. Save the page.

The renderer finds the wrapper, parses the JSON, hides the raw data, and replaces it with a styled statblock.

Correct `.monster` Wrapper

Use this first. It is the cleanest option because `<template>` content stays hidden before the renderer runs.

```
<template class="dnd-monster-data">
{
  "...": "paste the full .monster JSON here"
}
</template>
```

For a real `.monster` file, paste the **entire file contents** between the opening and closing tags:

```
<template class="dnd-monster-data">
{
  "name": "Ancient Red Dragon",
  "size": "gargantuan",
  "type": "dragon",
  "...": "the rest of the saved Tetra-cube .monster JSON"
}
</template>
```

Important: when pasting into BookStack, paste only the HTML snippet. Do not paste the Markdown code fence lines.

Most Compatible Wrapper

If BookStack strips `<template>` tags, use a normal `<div>` instead.

```
<div class="dnd-monster-data">
{
  "...": "paste the full .monster JSON here"
}
</div>
```

This is slightly less clean before render, but the CSS hides it and the renderer still replaces it with the statblock.

Recommended `.monster` Wrapper With Useful Options

This is the pattern I would use for most BookStack pages while building/testing monsters:

```
<template
  class="dnd-monster-data"
  data-copy-buttons="true"
  data-show-source="true"
  data-source-label="Original .monster JSON">
{
  "...": "paste the full .monster JSON here"
}
</template>
```

What those attributes do:

- `data-copy-buttons="true"` adds small copy controls beside attack bonuses, save DCs, and dice formulas.
- `data-show-source="true"` adds a collapsed source/debug block with the original JSON.
- `data-source-label="Original .monster JSON"` changes the label on that collapsed source block.

After the monster is working, you can remove `data-show-source="true"` if you do not want the source/debug disclosure visible.

Layout Options

Add these attributes to the wrapper.

```
<template
  class="dnd-monster-data"
  data-layout="one-column">
{
  "...": "paste the full .monster JSON here"
}
</template>
```

Available layout values:

- `data-layout="one-column"`
  Forces one-column output. Good for narrow BookStack pages or long monster actions.
- `data-layout="two-column"`
  Forces two-column output on desktop. Good for large monsters or boss statblocks.
- `data-layout="wide"`
  Makes the block wider.
- `data-layout="full"`
  Lets the block use the full available content width.

Compact example:

```
<template
  class="dnd-monster-data"
  data-layout="one-column"
  data-compact="true">
{
  "...": "paste the full .monster JSON here"
}
</template>
```

Boss monster example:

```
<template
  class="dnd-monster-data"
  data-layout="two-column"
  data-copy-buttons="true">
{
  "...": "paste the full .monster JSON here"
}
</template>
```

Tetra-cube `.monster` Workflow

1. Build or edit the monster at Tetra-cube.

2. Click **Save Statblock**.
3. Open the downloaded `.monster` file in a text editor.
4. Copy all of the JSON.
5. Paste it inside a `dnd-monster-data` wrapper.
6. Save the BookStack page.

Do not paste:

- The visible statblock text from Tetra-cube.
- A local file path to the `.monster` file.
- An attachment URL.
- A `<script type="application/json">` block.

The renderer expects inline JSON inside:

```
<template class="dnd-monster-data">
...
</template>
```

or:

```
<div class="dnd-monster-data">
...
</div>
```

Full Example: `.monster`-Style Block

This is a shortened Tetra-cube-style example. A real `.monster` file will usually have more fields.

```
<template
  class="dnd-monster-data"
  data-copy-buttons="true"
  data-show-source="true">
{
  "name": "Emberglass Drake",
  "size": "medium",
  "type": "dragon",
  "alignment": "chaotic neutral",
  "armorClass": 15,
  "hpText": "76 (9d8 + 36)",
  "speed": 30,
  "flySpeed": 60,
  "strPoints": 18,
  "dexPoints": 14,
  "conPoints": 18,
  "intPoints": 8,
  "wisPoints": 12,
  "chaPoints": 15,
  "cr": "5",
  "skills": [
    { "name": "Perception", "note": "ex" },
    { "name": "Stealth" }
```

```
  ],
  "damagetypes": [
    { "name": "fire", "type": "i" }
  ],
  "darkvision": 60,
  "languages": [
    { "name": "Draconic", "speaks": true }
  ],
  "abilities": [
    {
      "name": "Heated Scales",
      "desc": "A creature that touches the drake or hits it with a melee attack while within 5
feet takes 3 (1d6) fire damage."
    }
  ],
  "actions": [
    {
      "name": "Bite",
      "desc": "_Melee Weapon Attack:_ [STR ATK] to hit, reach 5 ft., one target. _Hit:_ [STR
1D10] piercing damage plus [2D6] fire damage."
    },
    {
      "name": "Glassfire Breath (Recharge 5-6)",
      "desc": "The drake exhales fire in a 30-foot cone. Each creature in that area must make
a [CON SAVE] Dexterity saving throw, taking [8D6] fire damage on a failed save, or half as
much damage on a successful one."
    }
  ]
}
</template>
```

Hand-authored JSON Wrapper

Use `dnd-statblock-data` when you are writing your own simplified JSON instead of using a Tetra-cube `.monster` file.

```
<div class="dnd-statblock-data">
{
  "name": "Lantern-Haunt Wight",
  "meta": "Medium undead, lawful evil",
  "armorClass": "15 (ancient mail)",
  "hitPoints": "67 (9d8 + 27)",
  "speed": "30 ft.",
  "abilities": {
    "STR": "16 (+3)",
    "DEX": "12 (+1)",
    "CON": "16 (+3)",
    "INT": "11 (+0)",
    "WIS": "14 (+2)",
    "CHA": "15 (+2)"
  },
  "properties": [
    { "label": "Skills", "value": "Perception +4, Stealth +3" },
    { "label": "Damage Resistances", "value": "necrotic; bludgeoning, piercing, and slashing
from nonmagical attacks" },
```

```
    { "label": "Senses", "value": "darkvision 60 ft., passive Perception 14" },
    { "label": "Languages", "value": "Common plus one language it knew in life" },
    { "label": "Challenge", "value": "4 (1,100 XP)" }
  ],
  "sections": [
    {
      "title": "Actions",
      "features": [
        {
          "name": "Graveblade",
          "text": "Melee Weapon Attack: +5 to hit, reach 5 ft., one target. Hit: 8 (1d10 + 3)
slashing damage plus 7 (2d6) necrotic damage."
        }
      ]
    }
  ]
}
</div>
```

Multiple Hand-authored Monsters

Use `grid: true` and a `monsters` array.

```
<div class="dnd-statblock-data">
{
  "grid": true,
  "monsters": [
    {
      "name": "Ash Rat",
      "meta": "Tiny beast, unaligned",
      "armorClass": "12",
      "hitPoints": "7 (2d4 + 2)",
      "speed": "30 ft.",
      "abilities": {
        "STR": "4 (-3)",
        "DEX": "14 (+2)",
        "CON": "12 (+1)",
        "INT": "2 (-4)",
        "WIS": "10 (+0)",
        "CHA": "4 (-3)"
      },
      "properties": [
        { "label": "Senses", "value": "darkvision 30 ft., passive Perception 10" },
        { "label": "Challenge", "value": "1/8 (25 XP)" }
      ],
      "sections": [
        {
          "title": "Actions",
          "features": [
            {
              "name": "Bite",
              "text": "Melee Weapon Attack: +4 to hit, reach 5 ft., one target. Hit: 4 (1d4 +
2) piercing damage."
            }
          ]
```

```
        }
      ]
    },
    {
      "name": "Cinder Imp",
      "meta": "Tiny fiend, chaotic evil",
      "armorClass": "13",
      "hitPoints": "10 (3d4 + 3)",
      "speed": "20 ft., fly 30 ft.",
      "abilities": {
        "STR": "6 (-2)",
        "DEX": "16 (+3)",
        "CON": "12 (+1)",
        "INT": "10 (+0)",
        "WIS": "11 (+0)",
        "CHA": "13 (+1)"
      },
      "properties": [
        { "label": "Damage Immunities", "value": "fire" },
        { "label": "Senses", "value": "darkvision 60 ft., passive Perception 10" },
        { "label": "Challenge", "value": "1/2 (100 XP)" }
      ],
      "sections": [
        {
          "title": "Actions",
          "features": [
            {
              "name": "Spark Claw",
              "text": "Melee Weapon Attack: +5 to hit, reach 5 ft., one target. Hit: 5 (1d4 +
3) slashing damage plus 3 (1d6) fire damage."
            }
          ]
        }
      ]
    }
  ]
}
</div>
```

Supported Wrapper Attributes

Attribute	Values	Purpose
`data-layout`	`one-column`, `two-column`, `wide`, `full`	Changes statblock width/columns.
`data-compact`	`true`	Slightly tightens long blocks.
`data-copy-buttons`	`true`	Adds copy buttons for attacks, DCs, and dice.
`data-copy`	`true`	Alias for `data-copy-buttons="true"`.

`data-show-source`	`true`	Adds a collapsed original-source/debug block.
`data-source`	`true`	Alias for `data-show-source="true"`.
`data-source-label`	Any text	Changes the source disclosure label.

Tetra-cube Token Support

The renderer resolves common Tetra-cube bracket tokens:

Token	Meaning
`[MON]`	Short monster name.
`[MONS]`	Plural monster name.
`[STR]`, `[DEX + 2]`	Ability modifier, optionally adjusted.
`[STR ATK]`, `[DEX ATK - 1]`	Ability modifier plus proficiency.
`[WIS SAVE]`, `[CHA SAVE + 2]`	Save DC using 8 + ability modifier + proficiency.
`[STR 2D6]`, `[DEX 1D8 + 2]`	Average dice damage plus ability modifier.
`[3D6]`, `[2D10 + 4]`	Plain dice average and display formula.

Example:

```
{
  "name": "Bite",
  "desc": "_Melee Weapon Attack:_ [STR ATK] to hit, reach 10 ft., one target. _Hit:_ [STR
2D10] piercing damage."
}
```

Troubleshooting

The raw JSON appears on the page

The renderer CSS/JS is probably not installed in BookStack custom head content, or the wrapper class is wrong.

Check that you used one of these:

```
<template class="dnd-monster-data">
...
```

```
</template>
```

```
<div class="dnd-monster-data">
...
</div>
```

```
<div class="dnd-statblock-data">
...
</div>
```

BookStack will not save the wrapper

Use source/HTML mode. If BookStack strips `<template>`, use the `<div>` wrapper.

The page shows `Unexpected end of JSON input`

The pasted JSON is incomplete. Check for a missing closing `}` or `]`.

The block is too wide

Use:

```
data-layout="one-column"
```

The block is too long

Try:

```
data-layout="two-column"
```

or:

```
data-compact="true"
```

Best Default Snippet

Use this for most Tetra-cube `.monster` files:

```
<template
  class="dnd-monster-data"
  data-copy-buttons="true"
  data-show-source="true"
  data-source-label="Original .monster JSON">
{
  "...": "paste the full .monster JSON here"
```

```
}
</template>
```

If BookStack removes `<template>`, use this instead:

```
<div
  class="dnd-monster-data"
  data-copy-buttons="true"
  data-show-source="true"
  data-source-label="Original .monster JSON">
{
  "...": "paste the full .monster JSON here"
}
</div>
```

```
DND-MONSTER copy source
{"name":"Ancient Red Dragon","size":"gargantuan","type":"Dragon","tag":"","alignment":"chaotic e
```

Revision #19
Created 2026-06-23 18:31:05 UTC by Jax SN Johnson
Updated 2026-06-24 18:11:51 UTC by Jax SN Johnson