

ジェムリアTRPG ダイスボット 特記処理JSON記述マニュアル

Version 4.0 - Current Codebase

目次

- 1. [概要と基本構造](#)
 - 2. [cost \(コスト定義\)](#)
 - 3. [power_bonus \(威力ボーナス\)](#)
 - 4. [effects \(効果処理\)](#)
 - 5. [バフの命名規則と自動効果](#)
 - 6. [プラグイン効果のパラメータ](#)
 - 7. [実装パターン集](#)
-

1. 概要と基本構造

スキルデータの「特記処理」カラムには、以下の3つのルートキーを持つJSONオブジェクトを記述します。このJSONにより、スキルのコスト、威力補正、および発動時の効果(バフ、ダメージ、状態異常など)を制御します。

基本構造

```
{
  "cost": [ /* コスト定義配列 */ ],
  "power_bonus": [ /* 威力ボーナス定義配列 */ ],
  "effects": [ /* 効果定義配列 */ ]
}
```

2. cost (コスト定義)

スキル使用時に消費するステータスを定義します。

消費タイミング

- **即時発動スキル**: 「宣言」 ボタンを押して確定した瞬間に消費
- **通常スキル**: マッチ実行(ダイスロール)直前に消費

パラメータ

キー	必須	型	説明
type	○	String	消費するステータス名 ("MP", "FP", "HP", "亀裂" 等)
value	○	Integer	消費量

記述例

```
"cost": [  
  { "type": "MP", "value": 5 },  
  { "type": "FP", "value": 2 }  
]
```

3. power_bonus (威力ボーナス)

「威力計算」時および「マッチ実行」時に、コマンド(2d6+X)のXに加算される数値を定義します。

3.1 共通パラメータ

キー	必須	説明
source	○	参照する対象 ("self": 自分, "target": 相手)
param	○	参照するステータス名 ("HP", "MP", "亀裂", "破裂", "物理補正" 等)
operation	○	計算方法 (後述)
condition	△	適用条件オブジェクト (後述)
max_bonus	△	加算値の上限設定 (整数)

3.2 operation (演算タイプ)

operation	説明	必要な追加キー
FIXED	固定値を加算	value: 加算値
MULTIPLY	ステータス値 × 係数	value_per_param: 係数 (少数可)
PER_N_BONUS	ステータス値Nごとに加算 (値 // N) * value	per_N: Nの値 value: 1単位あたりの加算値
FIXED_IF_EXISTS	ステータス値が閾値以上なら加算	threshold: 閾値 (省略時1) value: 加算値

3.3 condition (条件指定)

conditionオブジェクトを記述すると、その条件を満たす場合のみボーナスが適用されます。

キー	説明	値の例
source	参照元	"self", "target", "skill" (自分のタグ), "target_skill" (相手のタグ)
param	パラメータ	ステータス名 または "tags" (タグ配列)
operator	比較演算子	"CONTAINS" (含む), "GTE" (≥), "LTE" (≤), "GT" (>), "LT" (<), "EQUALS" (==)
value	比較値	数値 または 文字列

記述例

相手の亀裂1につき威力+1、最大3:

```
"power_bonus": [
  {
    "source": "target",
    "param": "亀裂",
    "operation": "PER_N_BONUS",
    "per_N": 1,
    "value": 1,
    "max_bonus": 3
  }
]
```

4. effects (効果処理)

バフ付与、状態異常、追加ダメージなどを定義します。

4.1 基本構成

```
{
  "timing": "発動タイミング",
  "type": "効果タイプ",
  "target": "対象",
  "condition": { /* 条件オブジェクト(power_bonusと同じ) */ },
  /* 効果タイプごとのパラメータ */
}
```

4.2 timing (発動タイミング)

現在のsocket_battle.pyの実装順序に基づきます。

タイミング	説明	重要仕様
PRE_MATCH	マッチ実行直前	自己バフや即時発動スキル用
WIN	勝利時	攻撃側の場合、HITより先に処理されます これにより「勝利時にバフを得て、そのバフを命中時効果に乗せる」ことが可能です
HIT	命中時 / 成功時	攻撃側: 命中時(一方攻撃含む) 防御側: 回避・防御成功時
LOSE	敗北時	マッチに負けた時

タイミング	説明	重要仕様
UNOPPOSED	一方攻撃時	相手が防御不能時。HITと並行して処理されます
END_MATCH	終了時	引き分けや、勝敗に関わらない終了処理

4.3 type (効果タイプ) 一覧

タイプ	説明	必須/オプションキー
APPLY_STATE	ステータスの増減 (HP, FP, 亀裂など)	state_name: ステータス名 value: 変動値 (負数で減少)
APPLY_BUFF	バフの付与	buff_name: バフ名 (※命名規則あり) lasting: 持続R数 (999で永続) delay: 遅延R数
REMOVE_BUFF	バフの削除	buff_name: 削除するバフ名
SET_STATUS	ステータス固定値化	state_name: ステータス名 value: 設定する値
CUSTOM_DAMAGE	固定ダメージ追加	value: ダメージ量
MODIFY_ROLL	ロール結果の修正 (ログ表示のみ)	value: 修正値
CUSTOM_EFFECT	プラグイン効果の発動	value: 効果ID ("亀裂崩壊", "破裂爆発" 等) ※各プラグインのパラメータが必要
FORCE_UNOPPOSED	一方攻撃化	なし (PRE_MATCHで使用)

5. バフの命名規則と自動効果

`manager/buff_catalog.py`に定義された命名規則に従ってバフ名を付けることで、特殊な効果が自動的に適用されます。**JSON側に複雑なロジックを書かずとも、名前だけで機能します。**

5.1 威力アップ系

- **パターン:** `[任意の名前]_Atk[数値]`
- **効果:** 自分が「攻撃」タグを持つスキルを使用した際、威力を`[数値]`加算します
- **例:** 猛攻の輝き_Atk5

5.2 守備力アップ系

- **パターン:** `[任意の名前]_Def[数値]`
- **効果:** 自分が「守備」タグを持つスキルを使用した際、威力を`[数値]`加算します

5.3 亀裂付与ボーナス (永続/ラウンド持続)

- **パターン:** `[任意の名前]_Crack[数値]`
- **効果:** 自分が相手に「亀裂」を与える際、その付与量を`[数値]`増やします
- **消費:** **されません**(持続時間が切れるまで残ります)

5.4 亀裂付与ボーナス (1回使い切り)

- **パターン:** `[任意の名前]_CrackOnce[数値]`
 - **効果:** 自分が相手に「亀裂」を与える際、その付与量を`[数値]`増やします
 - **消費:** **効果が発動した(亀裂を与えた)直後に、自動的に削除されます**
 - **注意:** JSONに`REMOVE_BUFF`を書く必要はありません。名前をこれにするだけで自動消費されます
-

6. プラグイン効果のパラメータ

`CUSTOM_EFFECT`の`value`に指定するIDに応じたパラメータ記述が必要です。

6.1 亀裂崩壊 (`fissure.py`)

A. ダメージモード

対象の亀裂を全て消費し、ダメージに変換します。

- `value`: "亀裂崩壊_DAMAGE"
- `damage_per_fissure`: 亀裂1あたりのダメージ量 (Int)

B. 誘発モード

対象の亀裂を消費し、別の効果(破裂爆発など)を複数回発動させます。

- `value`: "亀裂崩壊"
- `data`:
 - `"cost_per_trigger"`: 1回の誘発に必要な亀裂数 (デフォルト5)
 - `"triggered_effect"`: 誘発する効果名 (デフォルト"破裂爆発")
 - `"max_triggers"`: 最大誘発回数 (0で無制限)
 - `"trigger_ratio"`: 誘発先に渡す倍率補正

6.2 破裂爆発 (`burst.py`)

対象の「破裂」ステータスに基づいてダメージを与えます。

- `value`: "破裂爆発"
 - `rupture_remainder_ratio`: 爆発後に残す破裂の割合 (0.0 ~ 1.0)
 - 0.5なら半分残る。記述がない場合は0(全消費)
 - `fixed_damage`: 破裂消費なしで与える固定ダメージがあれば指定
-

7. 実装パターン集

ケースA: 次の一撃だけ亀裂付与+1 (突き崩す)

バフ名を `_CrackOnce1` にすることで、命中時(亀裂付与時)にボーナスが適用され、即座にバフが消滅します。

```
{
  "cost": [{ "type": "FP", "value": 2 }],
  "effects": [
    {
      "timing": "PRE_MATCH",
      "type": "APPLY_BUFF",
      "target": "self",
      "buff_name": "突き崩す_CrackOnce1",
      "lasting": 999
    }
  ]
}
```

ケースB: 勝利時にバフを得て、即座にその攻撃に乗せる (そこが弱い?)

`WIN` でバフ付与 → `HIT` で亀裂付与の順で記述します。(`_CrackOnce` を使っても良いですが、ラウンド持続なら `_Crack` でOK)

```
{
  "effects": [
    {
      "timing": "WIN",
      "type": "APPLY_BUFF",
      "target": "self",
      "buff_name": "そこ弱_Crack1",
      "lasting": 1
    },
    {
      "timing": "HIT",
      "type": "APPLY_STATE",
      "target": "target",
      "state_name": "亀裂",
      "value": 2
    }
  ]
}
```


ケースC: 複雑な亀裂崩壊 (E-10)

相手の亀裂数に応じて威力が上がり、命中時に亀裂をダメージに変換する。

```
{
  "power_bonus": [
    {
      "source": "target",
      "param": "亀裂",
      "operation": "PER_N_BONUS",
      "per_N": 1,
      "value": 1,
      "max_bonus": 3
    }
  ],
  "effects": [
    {
      "timing": "HIT",
      "type": "CUSTOM_EFFECT",
      "target": "target",
      "value": "亀裂崩壊_DAMAGE",
      "damage_per_fissure": 4
    }
  ]
}
```

注意事項

- このマニュアルは現在のコードベース(`game_logic.py`, `socket_battle.py`, `buff_catalog.py`, `plugins/`)の実装に完全に準拠しています
- バフの命名規則を正しく使用することで、複雑なロジックをシンプルに実装できます
- プラグイン効果を使用する際は、各プラグインの仕様を確認してください

Document Version: 4.0 **Last Updated:** 2025