

JSON 記述方法マニュアル

概要

このマニュアルは、ゲームシステムの「特記処理」をJSON形式で記述するための完全ガイドです。スキルの威力ボーナスと付加効果を定義します。

基本構造

すべての特記処理JSONは、以下の2つのトップレベルキーを持つオブジェクトです：

```
{ "power_bonus": [ /* 威力ボーナスルール */ ], "effects": [ /* 効果ルール */ ] }
```

重要: どちらも配列です。複数のルールをカンマで区切って格納できます。ルールがない場合は空配列 [] を指定してください。

power_bonus (威力ボーナス)

スキル宣言時にチャットパレットの基本威力（ダイス値）を増減させます。

キー一覧

 PDF出力

キー	型	必須	説明
source	文字列	✓	参照元："self"（自分）または"target"（対象）
param	文字列	✓	参照パラメータ名（例："MP"、"破裂"）
operation	文字列	✓	計算方法（詳細は下記）
value_per_param	数値	※	MULTIPLY時に必須。乗算値
value	数値	※	FIXED_IF_EXISTSまたはPER_N_BONUS時に必須
per_N	数値	※	PER_N_BONUS時に必須。「Nごと」のN値
max_bonus	数値	-	このルールの最大ボーナス上限

operation（計算方法）の種類

1. MULTIPLY（乗算）

```
{ "source": "self", "param": "MP", "operation": "MULTIPLY",
"value_per_param": 0.5 }
```

参照パラメータの値にvalue_per_paramを乗算します。小数対応。

2. FIXED_IF_EXISTS（固定値）

```
{ "source": "target", "param": "毒", "operation": "FIXED_IF_EXISTS",
"value": 5 }
```

参照パラメータが1以上なら、固定値valueを加算します。

3. PER_N_BONUS（Nごと）

```
{ "source": "target", "param": "破裂", "operation": "PER_N_BONUS", "per_N": 3,
"value": 1, "max_bonus": 3 }
```

参照パラメータが**Nごと**にvalueを加算します。最大値制限可能。

💡 使用例

「対象の破裂3につき基礎威力+1（最大3）」

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

effects (効果配列)

マッチ中やラウンド終了時に特定のタイミングで特定の処理を実行します。

キー一覧

キー	型	必須	説明
timing	文字列	✓	発動タイミング (下記参照)
type	文字列	✓	効果タイプ (下記参照)
target	文字列	✓	対象 : "self" または "target"
condition	オブジェクト	-	発動条件 (任意。指定なしで無条件発動)
その他	-	type次第	各typeが必要とする追加キー

timing (発動タイミング)

timing	発動場面
"PRE_MATCH"	スキル宣言時
"OPPONENT_ROLL_MOD"	ダイスロール直後
"UNOPPOSED"	一方攻撃成立時
"WIN"	マッチ勝利時 (ダイスロール勝ち)
"HIT"	的中時 (WIN または UNOPPOSED)
"LOSE"	マッチ敗北時
"END_MATCH"	マッチ終了時 (主に引き分け)
"END_ROUND"	ラウンド終了時

type (効果タイプ)

1. APPLY_STATE (状態付与)

```
{ "timing": "HIT", "type": "APPLY_STATE", "target": "target",
  "state_name": "出血", "value": 2 }
```

states配列を操作します。valueで増減値を指定（負数で減少）。

2. APPLY_BUFF (バフ追加)

```
{ "timing": "PRE_MATCH", "type": "APPLY_BUFF", "target": "self", "value":
  "挑発中" }
```

special_buffs配列にバフ名を追加します。

3. REMOVE_BUFF (バフ削除)

```
{ "timing": "END_MATCH", "type": "REMOVE_BUFF", "target": "self", "value":
  "挑発中" }
```

special_buffs配列からバフ名を削除します。

4. MODIFY_ROLL (ダイス修正)

```
{ "timing": "OPPONENT_ROLL_MOD", "type": "MODIFY_ROLL", "target":
  "target", "value": 3 }
```

相手のダイス結果を指定値だけ修正します（負数で減少）。

5. DAMAGE_BONUS (ダメージ追加)

```
{ "timing": "HIT", "type": "DAMAGE_BONUS", "target": "self", "value": 5 }
```

スキルダメージとは別に、固定ダメージを追加。

6. CUSTOM_EFFECT (カスタム処理)

```
{ "timing": "HIT", "type": "CUSTOM_EFFECT", "target": "target", "value":
  "破裂爆発", "rupture_remainder_ratio": 0.5 }
```

Python側の専用関数を呼び出します。valueで関数を指定。



CUSTOM_EFFECT の各関数

「破裂爆発」

対象の破裂値に基づくダメージを与え、破裂値を修正します。

```
{ "timing": "HIT", "type": "CUSTOM_EFFECT", "value": "破裂爆発",
"rupture_remainder_ratio": 0.5 }
```

パラメータ:

- `rupture_remainder_ratio` (数値, 0~1) : 爆発後に残る破裂の割合
 - 0.5 : 破裂が半分になる
 - 1 : 破裂が消費されない
 - 0または省略 : 破裂は0になる (全消費)

「亀裂崩壊」

対象の亀裂値をコストとして、破裂爆発を複数回実行します。

```
{ "timing": "HIT", "type": "CUSTOM_EFFECT", "value": "亀裂崩壊",
"cost_per_burst": 3 }
```

パラメータ:

- `cost_per_burst` (数値) : 1回の爆発に必要な亀裂コスト
 - 省略時 : デフォルト値5が使用される

「APPLY_SKILL_DAMAGE AGAIN」

そのスキルの基礎ダメージをもう一度与えます。

```
{ "timing": "HIT", "type": "CUSTOM_EFFECT", "value":
"APPLY_SKILL_DAMAGE AGAIN" }
```

用途: ダメージ2回付与スキル

パラメータ: なし

「出血氾濫」

Python側にハードコードされた出血氾濫のロジックを呼び出します。

```
{ "timing": "HIT", "type": "CUSTOM_EFFECT", "value": "出血氾濫" }
```

パラメータ: なし



condition (発動条件)

effectsオブジェクトにconditionを追加すると、効果に発動条件を設定できます。conditionがない場合は無条件発動です。

キー一覧

キー	型	必須	説明
source	文字列	✓	参照元："self"、"target"、"target_skill"
param	文字列	✓	参照パラメータ名
operator	文字列	✓	比較演算子（下記参照）
value	数値/文字列	✓	比較値

operator (比較演算子)

operator	意味
"GTE"	≥ (以上)
"LTE"	≤ (以下)
"GT"	> (より大きい)
"LT"	< (より小さい)
"EQUALS"	== (等しい)
"CONTAINS"	配列が値を含む (例: tags配列に"守備"が含まれるか)

💡 使用例

「自分の速度が8以上なら発動」

```
"condition": { "source": "self", "param": "速度", "operator": "GTE",  
"value": 8 }
```

「対象が守備スキルを使用した場合のみ」

```
"condition": { "source": "target_skill", "param": "tags", "operator":  
"CONTAINS", "value": "守備" }
```



実践サンプル集

サンプル1：Ps-04 「乱斬」

効果: 的中時に出血2を付与。自分の速度8以上なら、ダメージをもう一度与える

JSON コード

```
{
  "power_bonus": [],
  "effects": [
    {
      "timing": "HIT",
      "type": "APPLY_STATE",
      "target": "target",
      "state_name": "出血",
      "value": 2
    },
    {
      "timing": "HIT",
      "type": "CUSTOM_EFFECT",
      "target": "self",
      "value": "APPLY_SKILL_DAMAGE AGAIN",
      "condition": {
        "source": "self",
        "param": "速度",
        "operator": "GTE",
        "value": 8
      }
    }
  ]
}
```

サンプル2：E-13 「蔓延るがよい」

効果: 対象が守備スキル使用時にそのスキル威力+3。的中時に荊棘飛散

JSON コード

```
{
  "power_bonus": [],
  "effects": [
    {
      "timing": "OPPONENT_ROLL_MOD",
      "type": "MODIFY_ROLL",
      "target": "target",
      "value": 3,
      "condition": {
        "source": "target_skill",
        "param": "tags",
        "operator": "CONTAINS",
        "value": "守備"
      }
    },
    {
      "timing": "HIT",
      "type": "CUSTOM_EFFECT",
      "target": "target",
      "value": "荊棘飛散"
    }
  ]
}
```

サンプル3：Pb-05 「圧殺」

効果: 対象の破裂3につき威力+1（最大3）。的中時に破裂爆発（破裂が半分になる版と通常版の2回）

JSON コード

```
{
  "power_bonus": [
    {
      "source": "target",
      "param": "破裂",
      "operation": "PER_N_BONUS",
      "per_N": 3,
      "value": 1,
      "max_bonus": 3
    }
  ],
  "effects": [
    {
      "timing": "HIT",
      "type": "CUSTOM_EFFECT",
      "target": "target",
      "value": "破裂爆発",
      "rupture_remainder_ratio": 0.5
    },
    {
      "timing": "HIT",
      "type": "CUSTOM_EFFECT",
      "target": "target",
      "value": "破裂爆発"
    }
  ]
}
```

✓ 記述時のチェックリスト

- JSONの開き括弧 { と閉じ括弧 } が対応している
- 配列の開き括弧 [と閉じ括弧] が対応している
- キー名がダブルクオーテーション ”で囲まれている
- 文字列値もダブルクオーテーション ”で囲まれている
- 各要素の最後のカンマに注意（最後の要素にはカンマなし）
- source と target の指定が正しい（"self" または "target"）
- operation と operator を混同していない
- 必須パラメータが全て指定されている
- 型が正しい（数値は ” で囲まない）

このマニュアルはフェーズ3のデータ入力作業用です。

最終更新：2025年11月13日 | バージョン：1.0