

**NAME**

`gma go roll` – GMA die roller CLI application (Go version)

**SYNOPSIS**

(If using the full GMA core tool suite)

```
gma go roll [options as described below...]
```

(Otherwise)

```
roll -h
```

```
roll -help
```

```
roll -syntax
```

```
roll [-dice string] [-json] [-seed int]
```

**DESCRIPTION**

`Roll` provides a command-line utility that rolls dice using the GMA dice library without requiring a functioning GMA game server to roll the dice for you.

It can be used interactively by users or it can be embedded into scripts or other back-end services.

The full description of what can go into a die roll expression string is documented in `gma-dice-syntax(7)`.

**OPTIONS**

Options may be introduced with either one or two hyphens (e.g., `-json` or `--json`). Options which take parameter values may have the value separated from the option name by a space or an equals sign (e.g., `-seed=2323` or `-seed 2323`), except for boolean flags which may be given alone (e.g., `-json`) to indicate that the option is set to “true” or may be given an explicit value which must be attached to the option with an equals sign (e.g., `-json=true` or `-json=false`).

`-dice string` Specify the die-roll expression to be rolled, such as “`3d6`”.

Multiple die-roll expressions may be given as the value to this option, separated from one another by semicolons (“`;`”). In this case each will be rolled in the sequence given. This allows a group of die rolls to be performed from a single seed value in a single invocation of the program.

If this is not given, `roll` will interactively prompt for die-roll expressions and roll each as they’re typed in. Typing a blank line repeats the previous expression. The program will exit on EOF.

`-help` Print a command option summary and exit.

`-json` Output the results as a JSON string instead of plain text.

`-seed int` Instead of using a random seed value, base the die roll results on the given value. The `int` value is a 64-bit integer expressed in decimal digits.

`-syntax` Print a summary of the die-roll expression syntax and exit. In interactive mode, this help text may be produced by typing “`help`” as the input line.

**OUTPUT FORMATS****Text Output**

The plain-text output format (the default) briefly summarizes each result and the die rolls which were made to reach that result. This output uses ANSI escape codes to add colors for emphasis.

In case a die roll specification yields three or more results (e.g., from a spec such as “`3d6+5 | repeat 20`”), a line of statistics will be printed, showing the number of results (N), population mean value ( $\mu$ ), standard deviation ( $\sigma$ ), median value (Md), mode value(s) (Mo), and the sum of all results in the set ( $\Sigma$ ).

**JSON Output**

If JSON output is requested, a single JSON object will be printed to standard output, with the following fields. Fields which are not relevant or have a zero value may be omitted entirely from the output.

**ResultSet** (*list of objects*)

The value is a list of JSON objects, each containing one set of results from the set of die-roll specifications. Each value is an object with the following fields:

**Title** (*string*)

The title (label) string, if any, specified by the user (this is the initial part of the die-roll spec, separated from the rest of the expression by an equals sign).

**Results** (*list of objects*)

This is a list of JSON objects, each describing one of the results from a single die-roll specification. Each element of the list is an object with the following fields:

**ResultSuppressed** (*bool*)

If true, this indicates that the results are not shown to the user. They may have been sent privately to the GM, for example,

**InvalidRequest** (*bool*)

If true, this indicates that the request could not be understood and thus no results have been generated.

**Result** (*int*)

The total result of the die-roll expression, truncated to an integer (rounded toward zero).

**Details** (*list of objects*)

This list describes the die-roll expression that led to the result, including subtotals and the value rolled for each individual die. Each element of the list is a JSON object with the following fields:

**Type** The type of value this element represents.

**Value** The value of the element, expressed as a string.

**Stats** (*object or null*)

If there are 2 or more results in the result set, this object will provide statistics about the result set's values. It contains the following fields:

**N** (*int*) The population size (number of results analyzed).

**Mean** (*float*)

The population mean (simple average).

**Median** (*float*)

The median value of the result set.

**Mode** (*list of ints*)

The mode value (the value which occurred most). If there's a tie for the most popular number, all of the most popular ones are listed.

**StdDev** (*float*)

The standard deviation of the values.

**Sum** (*int*)

The sum of all the values in the result set.

**Seed** (*int*)

The random-number seed used to generate this set of die rolls.

**SEE ALSO**

[dice\(3\)](#), [gma-roll\(6\)](#).

This program is analogous to, but has more features than, the Python-based `gma roll` program from GMA-Core.

**AUTHOR**

Steve Willoughby / steve@madscience.zone.

**BUGS**

The default seed chosen by the underlying die-rolling library is always 32 bits long, but that may change in the future, so don't rely on that fact.

**COPYRIGHT**

Part of the GMA software suite, copyright © 1992–2026 by Steven L. Willoughby, Aloha, Oregon, USA.  
All Rights Reserved. Distributed under BSD-3-Clause License.