

zen Keep

Serving D2 Mod Makers Since 2000

[Forums](#) [Downloads](#) [Knowledge Base](#)



[News Portal](#) < [Home](#) < [Knowledge Base](#)

CubeMain.txt

CubeMain.txt

Written by: [kingpin](#)

Written on: Fri Jan 12, 2007 5:22 pm

Last modified: Thu Sep 09, 2021 9:35 pm

Views: 36336

Description: by Nefarius & Myhruginoc, updated and restyled by Harvestwombs

Categories: [File Guides \(1.10 - 1.14\)](#)

OVERVIEW:

This file controls the recipes for the Horadric Cube

This file is using:

[ItemTypes.txt](#)

[Misc.txt](#)

[Armor.txt](#)

[Weapons.txt](#)

[SetItems.txt](#)

[UniquelItems.txt](#)

[Properties.txt](#)

[ItemStatCost.txt](#)

description: This column has no function, it only serves as a comment field to make it easier to find specific recipes.

enabled: Boolean, this controls if a recipe is active in-game, **1** stands for yes and **0** stands for no.

ladder: Boolean, **1** if only allowed in ladder on realms (TBD) (Note: Also works for single player, tcp/ip)

min diff: Minimum game difficulty to use recipe (**0** = **all difficulties**, **1** = Nightmare and hell, **2** = hell only).

version: Tells the game if this is a old classic, new classic or expansion recipe, **0**=old classic d2, **1**=new classic

d2 and 100 is expansion recipe.

**I am not sure if 1 is used in this file, but this column works the same way it works in all the other item related files, so it probably does.*

op: Replaces realtime field, can now select requirement for allowing recipe (See [Appendix A](#) for more details).

param: Replaces realtime field, can now select requirement for allowing recipe (See [Appendix A](#) for more details).

value: Replaces realtime field, can now select threshold for custom requirement (See [Appendix A](#) for more details).

class: You can use this to make a recipe class specific by using the class codes.

In example: ama, bar, pal, nec, sor, dru and ass.

numinputs: Controls the number of items you need to put into the cube for this recipe.

Note: If you are using the "qty" parameter in any inputs this number must include the sum of quantities.

inputs 1: through **input 7:** This controls the type of inputs used for the recipe, each cell stands for a different item. Also includes input parameters, (See [Appendix B](#) for more details).

input 1 - primary item of recipe, target of outputs for item transform recipes.

output: Defines the first recipe result item. Also includes output parameters, (See [Appendix B](#) for more details).

lvl: Controls the level of the item that is created in the first output field

Note this value force spawns the item at this specific level. Its also used in the formula for the next two fields.

plvl: This column uses a percent of the players level for the output level.

ilvl: This column uses a percent of the first input's level for the output level.

mod 1: through **mod 5:** Defines additional attributes affixed to the first output item.

- **mod #** - the property code used to specify an attribute.
- **mod # chance** - the chance in percent the property will be assigned.
- **mod # param** - parameter field for properties that use parameters.
- **mod # min** - minimum field for properties that specify a range of values.
- **mod # max** - maximum field for properties that specify a range of values.

To make property have a fixed value make mod # min and mod # max the same.

output b: Defines the second recipe result item. Also includes output parameters, please refer to the Cubemain Code Reference in the keeps resource section to read about these codes.

b lvl: Controls the level of the item that is created in the second output field

Note this value force spawns the item at this specific level. Its also used in the formula for the next two fields.

b plvl: This column uses a portion of the players level for the output level.

b ilvl: This column uses a portion of the first input's level for the output level.

b mod 1: through **b mod 5:** Defines additional attributes affixed to the second output item. These fields work the same way as the mod # fields for the first item.

output c: Defines the third recipe result item. Also includes output parameters, please refer to the Cubemain Code Reference in the keeps resource section to read about these codes.

c lvl: Controls the level of the item that is created in the third output field

Note this value force spawns the item at this specific level. Its also used in the formula for the next two

fields.

c plvl: This column uses a portion of the players level for the output level.

c ilvl: This column uses a portion of the first input's level for the output level.

c mod 1: through **c mod 5:** Defines additional attributes affixed to the third output item. These fields work the same way as the mod # fields for the first item.

***eol:** line terminator, must be filled if using MS Excel or similar spreadsheet that eats trailing tabs.

APPENDIX A:

By Joel, Updated by Myhrginoc

Note: stat is using the ID from itemstatscost.txt (ie strength has ID 0)

CODE: [Select all](#)

```
Cubemain Ops (1.10)

op      Skip recipe if ...
-----
1      DayOfMonth is less than Param or greater than Value
2      DayOfWeek != Value ( 1 = Sunday, ...)

~~~~~ 10519 ~~~~~ Stat.Accr

3      player stat(param) < value
4      player stat(param) > value
5      player stat(param) = value
6      player stat(param) ! value
```

APPENDIX B:

By Nefarius, Updated by HarvestWombs

For slightly more detail visit the [Cubemain Code Reference](#)

CODE: [Select all](#)

```
upg = For inputs only, Item is upgraded.
bas = Base item
exc = Exceptional
eli = Elite

usetype = Uses base type of input1 for output
useitem = Uses item from input 1

uns = Unsocket Item (destroys gems)
rem = Unsocket Item (removes gems)

reg = Regenerate? Seems to reroll the item
mod = Transfers properties from input1 to output item.
```

Link to this article: [Select all](#)

```
[url=https://d2mods.info/forum/kb/viewarticle?a=284]Knowledge Base - CubeMain.txt[/url]
```

Login