

Alchemy Advanced
Features
v1.0
JRoush

Contents

1	Settings Menu & Config File	2
2	Recipes	3
3	Loading & Saving Recipes - Alchemists Notebooks	3
4	Standard Recipes	3
5	Using Potions as Ingredients	4
6	Subrecipes	5
7	Using the same Ingredient more than once	5
8	Brewing in Batches	5
9	Apparatus & Skill Displays	6
10	Ingredient Filters	7
11	Ingredient Autofilters	8
12	Refiners	9
13	Effects of Alchemy Skill	9
14	Scripted Effects	9
15	Potion Weight	10
16	Potion Gold Value	10
17	Potion Effect Duration	10
18	Potion Effect Strength	12
18.1	Raw Effect Strength	12
18.2	Apparatus & Skill Bonuses	12
18.3	Final Duration and Magnitude	13

1 Settings Menu & Config File

Alchemy Advanced mod settings, shown in this document in SMALL CAPS, are saved with the savegame. Each time a savegame is loaded, however, Alchemy Advanced will override the saved values with any mod settings found in *Oblivion/Data/AlchemyAdvanced.ini*. This file may not exist, or may not specify a particular setting, in which case the setting(s) will keep their saved values. See the file itself for details.

Mod settings can be changed in-game using the Settings Menu - click the **Settings...** button on the main alchemy menu. The settings are grouped roughly by category, and listed by brief description. Mouse over the description to see the actual setting name. To change a setting, select it in the list and choose the appropriate action from the five buttons below the list panel. Numerical settings are generally adjusted with the + and - buttons on the right, while non-numerical settings are usually toggled through their possible values with the middle button. Many settings have a specific value that 'disables' the related feature, which can be set by clicking the **Disable** button. Nearly all settings have a 'default' value that can be set with the **Default** button.

The settings menu itself can be disabled. Its visibility is controlled by PARAM_SHOW_MENU - the default value of 1 makes the menu visible, and the 'disabled' value of 0 hides the menu. Note that changes to this setting only take effect after the entire alchemy menu is closed and reopened.

Mod settings can also be changed manually by opening the in-game console (press ~) and typing

Set RshAlchQuestAlchemy.SETTING_NAME to VALUE

This is especially useful for tweaking numerical settings without tediously clicking the + or - buttons repeatedly. Using the console to force settings to values not reachable in the settings menu is discouraged.

Note that there are a few settings that are hardcoded into the mod (e.g. number of ingredients per recipe) which cannot be changed by any means.

2 Recipes

Recipes are how Alchemy Advanced internally organizes information:

- A list of up to four ingredients. In Alchemy Advanced, an ‘ingredient’ may be an actual alchemical ingredient like Flax or Apple, a potion, or another recipe. For more details on the last two, see Sections 5 & 6.
- A list of which effects have been ‘refined’ (see Section 12) for each ingredient.
- A name for the recipe, which will be the name for the resulting potion. If no name is explicitly provided, the default is the name of the strongest effect, or ‘Empty Potion’ if no effects are present.
- (Optional) A maximum duration for the potions effects - see Section 17.

The main alchemy menu allows the player to edit and brew a single recipe.

3 Loading & Saving Recipes - Alchemists Notebooks

A blank recipe is created when the alchemy menu is first opened. The **Save** and **Load** buttons on the bottom allow the player access to a collection of stored recipes that are saved with the players savegame.

This Saving and Loading scheme is a bit different from most. There are *two* ways to load recipe: **Load Copy** and **Edit**. **Edit** loads the actual recipe - any changes made to it will *immediately* alter the original. There is no need to click Save again after Editing a stored recipe, and doing so will have no effect. **Load Copy** loads a *copy* of the saved recipe - any changes made will not affect the original. Clicking Save after loading a copy will simply save the copy.

A new type of apparatus, the Alchemists Notebook - has been introduced to determine how many saved recipes the player may access. Unlike other apparati, the strengths of all notebooks in the players inventory will add cumulatively. For example: if the player is carrying a novice Notebook (1 recipe) and an expert Notebook (10 recipes), they will be able to access a total of 11 saved recipes. If the player has more Recipes saved than they may access, the extra recipes are disabled in the loading menu until the player acquires more or larger Notebooks.

Alchemist’s Notebooks in strengths from Novice to Master will appear in random loot and vendor’s stocks along with other apparati. For the impatient, the novice version is available immediately at the Gilded Carafe.

4 Standard Recipes

Standard Recipes are a new type of item, similar to an Alchemist’s Notebook, but granting access to useful pre-generated recipes. These recipes are always available while the item is in the players inventory (they aren’t affected by the limits of Alchemist’s Notebooks). Standard Recipes cannot be edited in-game; they may only be loaded as copies. The player may, however, save a copy to their Alchemists Notebooks and edit that.

No Standard Recipes are included in Alchemy Advanced - only the mechanism for them, as a resource for other mods.

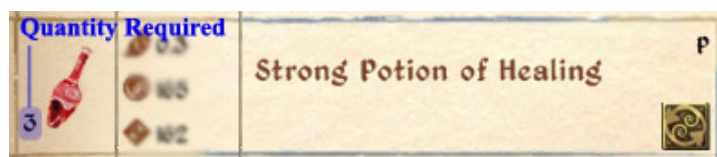
5 Using Potions as Ingredients

In addition to standard ingredient items (Bonemeal or Ectoplasm), potions may also be used as ‘ingredients’ in recipes. This includes potions bought or looted and potions previously brewed by the player. Just as with regular ingredients, the strength/value/rarity of the potion is irrelevant - only the order of the effects matters.

For the sake of balance, it generally takes more than one potion to fill a single ingredient slot:

$$\text{Potions per Slot} = (\text{Base Alchemy Skill}) / \text{RECIPE_PTN_INGRED_MULT}$$

. The default value for RECIPE_PTN_INGRED_MULT is 200 - at Base Alchemy Skill of 75, for example, it would take 3 ‘ingredient’ potions to brew a single instance of the recipe. Setting RECIPE_PTN_INGRED_MULT to -1 will entirely disable the use of potions as ingredients. For reference, the required quantity of an ingredient is shown below & to the left of its icon:



There are two main uses for this feature. The first is for higher level alchemists to ‘rebrew’ potions found as loot, making them more effective in the late game. Note that in general this will reduce their gold value, as there is a double standard in pricing for looted potions vs player-brewed potions.

The second use of this feature is to distill effects from several ingredients into an ‘intermediate’ potion, used as an ingredient to create a final, more powerful potion. For example, it is impossible using Oblivion+SI ingredients to create a poison with Paralyze, Damage Health, and all three elemental damage effects. It is relatively easy, however, to brew a few poisons with just three of these effects. By using these intermediary poisons as ingredients in another recipe, the mentioned five-effect poison becomes attainable:



As a shortcut to manually brewing each intermediary potion, it is possible to use a *Subrecipe* directly in an ingredient slot - see Section 6.

6 Subrecipes

A Subrecipe is a shortcut for manually brewing a potion and immediately using it as an ingredient in (see Section 5 on using potions as ingredients). When a parent recipe is brewed, any child recipes are automatically brewed first and their products are used as ingredients:



Subrecipes may themselves contain subrecipes, and so on. With enough such nested subrecipes (and raw ingredients), it is possible to obtain literally any combination of effects in the final result.

Saved and standard recipes are selected just like other ingredients from the ingredient selection menu. Only a copy of a saved recipe is used, so making changes won't affect the original. To edit or remove a subrecipe once it has been inserted, click on its name in the main alchemy menu. Recipes with subrecipes may be saved and loaded normally (see section 3.)

7 Using the same Ingredient more than once

In vanilla Oblivion, it is impossible to use an ingredient twice in the same recipe. In Alchemy Advanced, an ingredient or potion may be used in `RECIPE_INGRED_REP` slots. The default value is 1, the same as vanilla Oblivion. However, it generally takes more than one of a potion item to 'fill' a single ingredient slot - see Section 5.

This setting does NOT apply to subrecipes; each subrecipe is an independent copy which may or may not use the same ingredients as another subrecipe.

8 Brewing in Batches

A new quantity slider bar on the main menu allows potions to be brewed in quantities larger than one - should save wear & tear on the index finger. The player will receive the appropriate experience for *every* potion brewed, but only after the batch is finished. This means that any skill increases caused by a brewing a batch will not affect the strength of potions in the batch.

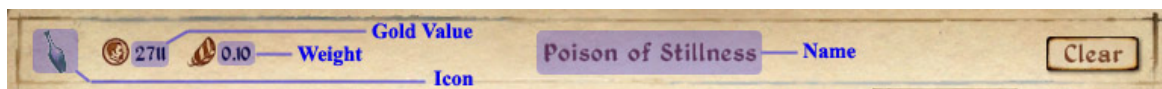
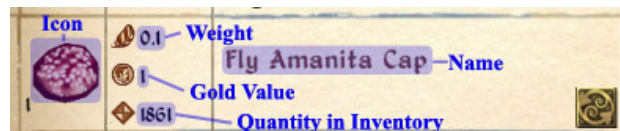
Batch size is capped by `RECIPE_BATCH_MAX`, which is 100 by default. This isn't for balance reasons so much as an effort to make fine-tuning the quantity slider less arduous. The quantity slider can't be disabled, mostly because I can't see why anyone would want to.

9 Apparatus & Skill Displays

For each apparatus type in the players inventory, the equipped apparatus' strength is now shown near the apparatus icon. For Alchemists Notebooks and Refiners, the current number of saved recipes and refined effects is also shown. Mouse over an apparatus icon to display the name of the apparatus being used. The player's base and effective (modified by luck) alchemy skills are shown at the center of the apparatus row - the base value is in parenthesis.



The weight and gold value of each ingredient is now shown to the left the ingredient name, and the weight, gold value, and icon of the resulting potion is shown to the left of the recipe name.



10 Ingredient Filters

The ingredient selection menu allows the player to pick an ingredient for the current recipe. For an ingredient to be shown, the player must have it in their inventory (obviously), and it must pass the current *filter*.

An ingredient filter is a list of conditions combined with a boolean operator:

- *AND* - every condition in the list must be met
- *OR* - at least one of the conditions must be met
- *XOR* - an odd number of the conditions must be met (useful for “either-or, but not both” situations)
- *NAND* - at least one of the conditions must not be met
- *NOR* - none of the conditions may be met
- *NXOR* - an even number of the conditions must be met

To change the operator, select it in the list and click the “**OR/AND/XOR/NO/NAND/NXOR**” toggle button.

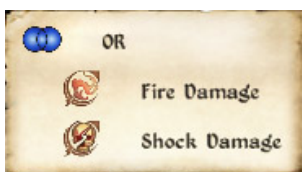
Generally the conditions are that the ingredient has some particular effect(s). For example, a “Paralyze” condition is met if the ingredient has a “Paralyze” effect visible to the player at their current skill level. Another condition is the *absence* of a particular effect, e.g. a “NOT Paralyze” condition is met if the ingredient *doesn’t* have a visible “Paralyze” effect. To switch an effect condition between ‘Standard’ and ‘NOT’, click the “**NOT/STD**” toggle button. Some effects (e.g. “Fortify Attribute” or “Script Effect”) take Actor Value or Script parameters. These can be specified by clicking on **Attr/Skill...** or **Script...** See section 14 for details on how Alchemy Advances determines whether two “Script Effects” match.

It is also possible for a condition to be a *Subfilter* - another boolean operator with its own list of conditions. In this case the subfilter ‘condition’ is met if the ingredient passes the subfilter.

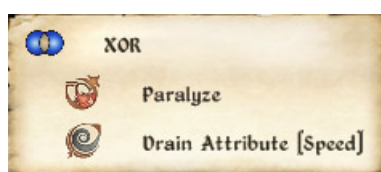
Examples:



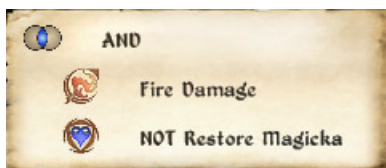
Fire Damage AND Damage Health effects



Fire Damage OR Shock Damage effect



Either Paralyze or Drain Speed effect, but not both



Fire Damage but no Restore Magicka effect



Restore Fatigue but no Restore Magicka OR vise-versa OR neither



Paralyze AND at least one of the elemental damage effects

To edit the current ingredient filter, click the **“Filter”** button in the ingredient selection menu. Note that an empty filter - one with no conditions - will pass all ingredients, regardless of what operator is chosen.

11 Ingredient Autofilters

Two automatic filters are available:

Match Recipe shows only ingredients with at least one visible effect in common with the ingredients already in the recipe. This is the vanilla Oblivion filtering scheme. To generate this filter, click an ingredient slot and then the **Match Recipe** button.

Match Ingredient shows only ingredients that (1) have all effects from the selected ingredient that appear on the final potion, and (2) have no effects from other ingredients in the recipe that don't appear in the final potion. Useful for finding substitutes for a depleted ingredient which won't alter the final potion. To generate this filter, click the ingredient slot containing a depleted ingredient, and then the **Match Ingrid** button.

The `INGSEL_AUTOFILTER` setting determines what, if anything, is done to the ingredient filter each time the ingredient selection menu is opened. The default value is -1, meaning the filter isn't changed. A value of 0 clears the filter. A value of 1 applies the Match Recipe autofilter (this is the vanilla Oblivion behavior).

12 Refiners

Refiners are a new type of apparatus that allow the player to selectively mask effects from ingredients. One use is to remove benign effects from would-be poisons. Master alchemists can also use them to make single-ingredient potions using the second, third, etc, effects from an ingredient. Refiners are *extremely* handy.

To mask an effect on an ingredient, mouse over the ‘magic effects’ icon near the ingredient name and click the desired effect on the popup list. While masked, an effect will not contribute to the final potion. Clicking the effect again will unmask it.



Stronger refiners allow more effects to be refined in a single recipe. The refining state of each ingredient is stored in the recipe when saved/loaded. Refiners in strengths from Novice to Master will appear in random loot and vendor’s stocks along with other apparati. For the impatient, the novice version is available immediately at the Gilded Carafe.

13 Effects of Alchemy Skill

For base alchemy skills below master level, the Oblivion game settings *iMagicMaxPotionsNovice* ... *iMagicMaxPotionsMaster* determine how many effects are visible to the player and used in brewing potions. The default values start at 1 and increase by 1 for each mastery rank to a max of 5 at master level. With the default master skill level being 100, this means that the player can see one additional effect for every 25 skill levels.

Alchemy Advanced also extends this for skill levels over 100 (only reachable using Elys’ Skill Uncapper). The player can see one addition effect for every `INGRED_EFF_SKILL` skill levels above master. The default value is 25, meaning that by default eight effects would be visible at level 175. The complete formula for base alchemy skills greater master level is:

$$\text{Effects Visible} = \lfloor iMagicMaxPotionsMaster + \text{INGRED_EFF_SKILL} * (\text{Skill} - iMagicMaxPotionsMaster) \rfloor$$

To make *all* effects visible at master level, set `INGRED_EFF_SKILL` to zero (recommended for anyone not using Elys’ Uncapper).

14 Scripted Effects

Two “Script Effects” are considered to be identical (for the purposes of brewing and filtering) if they use the same script. Note that different ingredients might have effects with different names and hostility states that use the same script. When brewing a potion with a scripted effect, the name and hostility will be copied from the first occurrence of that script in the ingredients list. All script effects on the final potion will have magnitude and duration equal to 1, and do not contribute to the gold value.

Depending on how a particular effect was scripted, it may have unpredictable effects when transferred to a brewed potion. Thus, while Alchemy Advanced makes it *possible* to brew potions with script effects, it’s generally not very useful.

15 Potion Weight

The weight of a brewed potion is the average weight of its ingredients (as in the vanilla system), but constrained between POTION_WEIGHT_MIN and POTION_WEIGHT_MAX:

$$(\text{Final Weight}) W = \begin{cases} \text{POTION_WEIGHT_MIN} & AvgWeight \leq \text{POTION_WEIGHT_MIN} \\ \text{POTION_WEIGHT_MAX} & AvgWeight \geq \text{POTION_WEIGHT_MAX} \\ AvgWeight & \text{otherwise} \end{cases}$$

The default values are both 0.1, meaning that every brewed potion weighs 0.1 lbs. Setting POTION_WEIGHT_MIN to 0.0 effectively disables the minimum, and setting POTION_WEIGHT_MAX to -1.0 (the internal flag for infinity) disables the maximum.

16 Potion Gold Value

The game setting *fPotionGoldValueMult* (default value .45) is used to automatically calculate the gold value of *all* potions, including those found as loot or purchased from vendors. For a brewed potion, Alchemy Advanced also uses the parameter POTION_GOLD_MULT and the total magicka cost of the potions effects to determine its ‘raw’ gold value:

$$(\text{Raw Gold Value}) G_{Raw} = fPotionGoldValueMult * \text{POTION_GOLD_MULT} * \sum_{Effects} (\text{Magicka cost of Effect})$$

The default value of POTION_GOLD_MULT is 0.5. Setting it to -1.0 will disable this feature and instead calculate the raw gold value using the formula from the vanilla Oblivion alchemy system.

The final gold value is the raw gold value constrained between POTION_GOLD_MIN and POTION_GOLD_MAX:

$$(\text{Final Gold Value}) G = \begin{cases} \text{POTION_GOLD_MIN} & G_{Raw} \leq \text{POTION_GOLD_MIN} \\ \text{POTION_GOLD_MAX} & G_{Raw} \geq \text{POTION_GOLD_MAX} \\ G_{Raw} & \text{otherwise} \end{cases}$$

Setting POTION_GOLD_MIN to 0.0 effectively disables the minimum, and setting POTION_GOLD_MAX to -1.0 (the internal flag for infinity) disables the maximum. The default values are 0.0 and -1.0 (both disabled). Common alternative settings might be 0.0 and 100.0 to approximate the vanilla potion price range, or 0.0 and 0.0 to make all player-brewed potions worthless.

17 Potions Effect Duration

For potion effects with both a magnitude and a duration (e.g. most effects), the magnitude M is related to the duration D by:

$$M = \text{POTION_EFFECT_DUR_MULT} * fPotionT1MagMult * fMagicDurMagBaseCostMult * D$$

The game settings *fPotionT1MagMult* and *fMagicDurMagBaseCostMult* have default values of 2.5 and 0.1, and *POTION_EFFECT_DUR_MULT* has a default value of 1.0. Thus, by default, an effect duration is four times its magnitude. This may be somewhat different for players using the vanilla potion strength formulas - for details, see section 18.

The player may choose instead specify a ‘maximum’ duration using the duration slider on the alchemy menu. Effects with both a magnitude and duration will be rescaled to this duration, with their magnitudes increased accordingly. Effects with only a duration will be truncated if they would normally last longer. This feature allows the player new flexibility with potions; they can be fast-acting and powerful, or long-lasting and weak, or anything in between. This is especially useful for poisons, allowing them to act much more quickly and powerfully. Note that, for balance, potion strength in general has been decreased (see section 18).

The duration slider is constrained between *POTION_EFFECT_DUR_MIN* and *POTION_EFFECT_DUR_MAX*, which default to 15 and 240 seconds, respectively. Setting either *POTION_EFFECT_DUR_MIN* or *POTION_EFFECT_DUR_MAX* to zero will disable this feature and hide the duration slider.

18 Potion Effect Strength

The Alchemy Advanced formulas for brewed potion strength are similar in spirit to the vanilla alchemy formulas, but very different (and much simpler) in detail. Which formulas are used is determined by `RECIPE_PTN_RSHEFFECTS` - the default value of 1 means using the new formulas, while a value of 0 reverts to the original formulas.

Compared to the vanilla alchemy formulas, these equations generally weaken effects with both a magnitude and duration and strengthen effects with only one or the other. Poisons are also somewhat stronger overall. The end result is that all desirable effects (e.g. benign effects on potions or hostile effects on poisons) now have more or less the same Apparatus & Skill Bonus, which was not even close to true in vanilla Oblivion. To make a long story short, the Alchemy Advanced strength formulas change the balance for brewed potions to be similar to that of player-made spells and enchantments.

18.1 Raw Effect Strength

Effect strengths are determined by three basic factors. The first is the ‘raw effect strength’, determined by how many ingredients the effect appears on, and where on each ingredient (first effect, second effect, etc). The formula for the raw strength S_{raw} of an effect is:

$$(\text{Slot Factor}) \chi_{slot} = \text{INGRED_EFF_SLOTBASE} + \text{INGRED_EFF_SLOTMULT} * \text{SlotNumber}$$

$$S_{Raw} = \text{INGRED_EFF_BASE} + \text{INGRED_EFF_MULT} * \sum \chi_{slot}$$

The summation is over every occurrence of the effect in every ingredient. This means that (unlike in vanilla Oblivion) potion effects get stronger as more ingredients with the same effects are added. It also means that each effect on an ingredient is more powerful than the following effect. The default parameter values are

<code>INGRED_EFF_BASE</code>	0.3
<code>INGRED_EFF_MULT</code>	0.35
<code>INGRED_EFF_SLOTBASE</code>	1.0
<code>INGRED_EFF_SLOTMULT</code>	-0.1

Some example raw strengths using the default settings:

First effect in a single-ingredient potion (using the master alchemy perk)	0.65
Fourth effect in a single-ingredient potion (using the master alchemy perk)	0.35
1 st effect in first ingredient, 3 rd effect in second ingredient	0.8
2 nd effect in all four ingredients	1.3

Setting `INGRED_EFF_SLOTMULT` to zero will give all effects on an ingredient an equal ‘slot factor’. Setting `INGRED_EFF_MULT` to zero will give *all* effects in the final potion the same ‘raw strength’.

18.2 Apparatus & Skill Bonuses

Most of the complexity (and bugs) in the vanilla alchemy formulas arose because they applied apparatus bonuses *after* calculating the duration and magnitude of each effect. Alchemy Advanced computes them beforehand as another strength factor S_{bonus} :

$$\begin{aligned}\chi_{mrt} &= fPotionMortPestleMult * (\text{Mortar Strength}) \\ \chi_{cal} &= (fMagicCostScale * fT1CM + fT1CD) * APP_CALC_MULT * (\text{Calcinator Strength}) \\ \chi_{RA} &= (fMagicCostScale * (fT1RM - fT1AM) + fT1RD - fT1AD) * APP_OTHER_MULT \\ S_{bonus} &= (EffectiveSkill + \chi_{mrt}) * \left(1 + \chi_{cal} + \chi_{RA} * \begin{cases} (\text{Retort Strength}) & \text{benign effect on potion} \\ -(\text{Retort Strength}) & \text{benign effect on poison} \\ -(\text{Alembic Strength}) & \text{hostile effect on potion} \\ (\text{Alembic Strength}) & \text{hostile effect on poison} \end{cases} \right)\end{aligned}$$

Where *EffectiveSkill* is the Luck-modified alchemy skill of the player. The default values of APP_CALC_MULT and APP_OTHER_MULT are 2.5 and 0.5, respectively. Here *fT1CM*, etc are shorthand for a series of game settings:

fT1CM	fPotionT1CalMagMult
fT1CD	fPotionT1CalDurMult
fT1RM	fPotionT1DurMagMult
fT1RD	fPotionT1DurDurMult
fT1AM	fPotionT1AleMagMult
fT1AD	fPotionT1AleDurMult

18.3 Final Duration and Magnitude

The final magnitude M and duration D of each effect on a brewed potion is determined by solving:

$$M^{fMagicCostScale} * D = \frac{S_{raw} * S_{bonus} * \text{POTION_EFFECT_MULT}}{(\text{Base Effect Cost}) * fMagicDurMagBaseCostMult}$$

Given:

- Effects with Base Effect Cost of zero (e.g. Script Effect) have $M = D = 1$
- Effects with no Duration have $D = 1$
- Effects with no Magnitude have $M = 1$
- Effects with both a magnitude and duration have (see Section 17):

$$M = \text{POTION_EFFECT_DUR_MULT} * fPotionT1MagMult * fMagicDurMagBaseCostMult * D$$

- If a ‘maximum’ duration was specified, $D \leq \text{‘maximum duration’}$ (see section 17)

The default value of POTION_EFFECT_MULT is 0.8. It can be adjusted to change the overall strength of all brewed potions - in particular, anyone not using Elys’ Skill Uncapper should consider increasing the value somewhat to make their potions useful in the late game.