

Scriptor Magicae 2.0 is here, ushering in a new age of power for words everywhere, at least on Minecraft version 1.21.1 (+TODO: others ready by video release) on NeoForge and Fabric. This video is a comprehensive overview of the mod as it is now, with attention brought to any changes from previous versions of Scriptor, as well as a couple of extra treats I'm releasing along with the new version.

Let's start at the most basic, welcome to the Player's Guide to Scriptor 2.0!

- Scriptor 2.0 Player's Guide
 - All the stuff a new or returning player might want to know

This should cover all the stuff a new or returning player might want to know about the mod.

- Spell structure
 - Varies by world
 - Contains one subject and any number of groups of actions and descriptors
 - Spells must be constructed with the correct structure and words or else they won't work at all

First off, let's cover how spells work. Each world has its own words and its own spell structure. The spell structure for a world will always allow for one subject and any number of actions and descriptors, grouped and separated by a particular word.

Subjects determine how a spell selects its target. This includes things like firing a projectile, targeting something in front of the caster, or even the caster themselves!

Actions decide what the spell will actually do, like healing, or teleporting their target, applying potion effects, or setting them ablaze. There are a lot of different actions, and a lot of them have multiple different applications!

And lastly are Descriptors! These can modify the spell as a whole or the action they're grouped with, and have a wide range of effects, from moving where the spell targets to reducing the cost to cast the spell, even adding colors, powering up spells, really the sky's the limit with these.

Spells must be constructed using the correct structure, or else they won't work at all!

- The Scriptionary
 - The Nominomist's Guide
 - A rough compilation of notes on the way magic works
 - Contains a few basic entries at the start of the game
 - Discovering new entries
 - Through your actions
 - Through notes

To help you keep track of things like spell structure and words, you might want to check out the Scriptionary, a sort-of-new item that gained a lot of functionality in Scriptor 2.0. This book is pretty simple to craft, by default only requiring [recipe]

Once you've crafted a Scriptionary, you can crack it open to take a look at some of the entries recorded within. Initially, you'll only have a few notes on casting spells and writing tomes, but you can unlock new entries by exploring the world, casting spells, interacting with entities, or redeeming notes you find around the world.

- Entries
 - Adding words at the writing desk

- Rubbing words from engravings and chalk
- Searching and viewing entries on the go

Next up are the entries to the actual dictionary part of the Scriptionary. This is where you can peruse recorded words, along with the meaning you noted down for them. To start with, this section will be empty, but there are a couple of ways you can add entries.

The first is by putting your Scriptionary into the top slot of a Writing Desk, then putting a spellbook or book into the other slot. This will let you select words to add to your Scriptionary directly, letting you write a brief description for them so you can keep track of what's what.

But that's only one way! When out exploring the world, you can also find engravings, or maybe you see a chalk circle in a friend's base. If you want to record a word from one of these, just use your Scriptionary on it and it'll take a rubbing of it, which you can assign a definition later at a Writing Desk.

Finally, though, there are Scraps. When you redeem a Scrap, if you have a Scriptionary in your inventory and the word written on your scrap hasn't already been recorded into it, you'll stuff the scrap into the entries section of your Scriptionary along with the description that was written on the scrap itself.

Once words are recorded, you can search through all the entries in your Scriptionary on the go, or put it in a Writing Desk with a Book and Quill or Spellbook and Quill to directly insert words into the books.

- Observations
 - We'll get to observations in a second

The last button you'll see in the Scriptionary is the Observations section. Initially, this won't actually be in your book, but we'll get to how to unlock it, and what it does, in just a second.

- How to discover new spells

Alright, so you know the basics of what a spell contains, and how to navigate your Scriptionary, but where do you actually get new spells? You can't record words if there aren't spells to get them from, right?

- Tomes
 - Contain a complete spell which can usually be cast immediately
 - Most players can pretty much get away with only using spells from tomes they find
 - Found in loot chests
 - Dropped from some bosses
 - Writing Desk
 - can click words to add them to your scriptionary

The first way to get new spells are Tomes. These contain a complete spell which can be cast immediately once revealed. Honestly, most players can pretty much get away with only using spells from tomes they find, since there are a wide range of useful spells out there already.

These tomes can be found in loot chests in a variety of dungeons, and as drops from some bosses, like the Wither and Ender Dragon.

As mentioned before, the Writing Desk is invaluable when it comes to recording words from these Tomes, making sure you copy down the words correctly every time.

- Engravings
 - Located underground and in villages

- Are a version of a spell engraved into the ground

Next up are Engravings. These can be found scattered around the world, engraved into stone in abandoned ritual sites and villages.

The engravings you find around the world will have their first word highlighted, which can be activated to cast the spell, after which they'll be burnt out for a bit. As mentioned before, you can use your Scriptionary on the words these contain to add them to your entries when you're out and about.

- Mobs
 - Some mobs, like Evokers and Villagers, will speak spells out loud when casting them.
 - Some words don't have corresponding spell effects, so they'll come across as gibberish
 - Observations
 - Recorded automatically to a scriptionary in your inventory when you hear a mob speak a complete spell

Another way to find new spells are mobs! Some mobs will cast spells when certain conditions are met, like Evokers when they're attacking a player and Nominomist Villagers when healing other villagers. That being said, some of the spells they cast don't have corresponding words yet, so they'll end up sounding a bit like gibberish.

When you hear a mob cast a spell like this, you'll automatically record it into the Observations section of your Scriptionary. I told you we'd get back to it!

Once you've recorded any spells like this, you can click the "Observations" button to read through any spells you've observed.

- Scraps
 - Contain single words, with a brief description of what the word does
 - Can be found in loot chests

Scraps, as mentioned before, contain single words, along with a brief description of what the word does. These can be found in loot chests scattered around the world, and can either be stuffed directly into your Scriptionary, or recorded into it at a Writing Desk.

- Inscribed Tools
 - Some mobs will carry items with equipment inscribed with a spell. Examining these items allows you to deconstruct the spells written on them.

One new way to find spells in Scriptor 2.0 are inscribed tools and armor! On occasion, you'll happen upon a mob holding a unique tool or wearing a piece of inscribed gear that contains a spell. These spells will usually apply specifically when the gear is used to attack, or cast on the attacker when they damage the wearer, but theoretically any spell could be inscribed on these bits of kit.

When you find one, you can examine its tooltip to see the spell contained within, to be inscribed onto new equipment or deconstructed to learn its components.

- Nominomist Villagers
 - Will trade some crafting resources, or trade down high-tier tomes into lower-tier ones
 - Will sometimes cast spells on other villagers, which can be observed to learn new spell components
 - May rarely trade you their notes
 - A nominomist will always have the same set of notes.

Also new to this version are Nominomist villagers! Out on your travels, you might happen upon one of these villagers with a Writing Desk in their home. By trading with them, you can get resources used to write new spellbooks, but after a few trades you may also unlock tome trades!

Nominomists will trade you Ancient Tomes they have laying around for higher-tier tomes, helping you get your hands on more common spells if you have a bunch of high-tier tomes from slaying bosses or late-game dungeon crawling.

That's not all, though! Nominomists can also sometimes be heard practicing magic of their own! Watch out and hang around while these guys are conducting their business and you might just hear one of their spells.

Lastly, you can also find rare Artifacts scattered around the world, in rare loot chests or chambers built specifically to protect them. When you locate these Artifacts, they'll have powerful spells inscribed on them, just check the tooltip!

- Casting spells
 - General rules
 - A spell will fizzle if its cost is higher than the limit of the chosen casting medium.
 - Cost can be reduced by some words
 - A spell will *always* fail to cast if words are out of order or spelled incorrectly. The language of magic has strict grammar and spelling rules.

Alright, so you've learned a bunch of places you can find spells, those are all well and good, but what if you want to do something new? How do you go about creating and casting spells of your own?

Well, spells have a couple of general rules:

A spell will fizzle if its cost is higher than the cost limit of the chosen casting medium. If a spell costs too much, you might need to remove some words from it or add one of a handful of special words that reduce spell cost, usually either by consuming a resource or meeting a condition.

Secondly, a spell will *always* fail to cast if its words are out of order or spelled incorrectly. The language of magic is strict, and if you don't follow its spelling and grammar rules, you won't get very far.

With those laid out, lets get to the actual methods you can use!

- Tomes
 - Spellbook and Quill
 - Writing Desk
 - allows you to click to insert words from your Scriptionary
 - Converting a written book
 - craft with a spell binder
 - can also craft spellbooks with mundane binders to convert back to normal books. If you want to, at least
 - Dying and obfuscation
 - tomes are available in all of the normal colors you'd expect from mc
 - crafting one with glow ink allows it to be obfuscated, if you want to share a tome with someone without revealing your secrets
 - book of books

First up is the classic, the Spellbook! These can obviously be found by uncovering ancient tomes, but making your own is pretty simple, too.

To start, you'll either want a Book and Quill or a Spellbook and Quill. These function mostly identically, though to use a normal written book as a spellbook you'll need to replace the binder by crafting the book with a spellbook binder.

To write these, you can either use the normal book writing interface, or the Writing Desk, a tool that, along with a Scriptionary, lets you pick and place words to help avoid spelling mistakes and the like.

When you have a Spellbook, casting with it is as easy as holding it in your hand and using it.

And if you ever want to convert a Spellbook back into a regular book, you can always craft it with a Mundane Binder.

Once you have a spellbook, you can also dye it to make identifying books easier, or obfuscate it by crafting the book with glow ink to keep others from reading it.

And last, but certainly not least, is the book of books. Spellbooks can kinda be hotbar hogs if you let them, but the book of books lets you combine a handful of books into one slot. Just hold a spellbook in your cursor and right click on the book of books to insert it, or right click with an empty cursor to take one out.

When you have a book of books in your hand, you can cast the spell within like normal, and cycle between spells by sneaking and scrolling your mouse wheel.

Alright, what's next?

- Artifacts
 - No cost limit
 - Will still incur a cooldown on the item itself
 - Don't trigger a player's casting cooldown
 - Cannot be created

Well, that'd be Artifacts. We went over how to discover them, but how do you make one yourself? That's the rub, you can't. These are powerful, with no limit on casting cost and the ability to bypass the player's casting cooldown, which means that if you find one, you should treasure it!

- Tool inscriptions
 - Can either cast normally on right click, or apply an effect when you damage an entity with it or are damaged while wearing it
 - Made by combining an item with a spellbook at an anvil

You know what you can do yourself, though? Inscribe spells onto tools!

And doing it is pretty intuitive, too! All you need to do is combine a tool with a Spellbook at an Anvil, and if you ever want to remove an inscription, you can pop the item into a Grindstone to do so.

While you probably want to inscribe your tool with a spell that casts on hit or your armor with one that goes off when you are damaged, you can technically inscribe any spell onto a piece of gear. If the spell is one of these other ones, just using your item is enough to cast it. Also a useful way to save hotbar slots, I suppose, though the cost limit for spells cast this way is a decent bit lower than casting through a Spellbook.

- Scrolls
 - Can be written at a writing desk

- Can only be cast once, but are cheaper to make than a tome and have a slightly higher cost limit by default

Well, if inscriptions have a lower ceiling for casting than Spellbooks, is there anything that can cast more powerful spells without requiring you to rewrite your whole spell?

One option is to cast using a Scroll. These are a good chunk cheaper than Spellbooks individually, but can only be used once each. That being said, they do also have a slightly higher casting limit, letting you cast spells that might be too powerful otherwise.

To make these, just craft up an unwritten scroll, then write a spell on it at a Writing Desk.

- Chalk
 - Can be written by renaming chalk for each glyph then writing out the spell by right clicking on the ground with the chalk corresponding to each word
 - Won't trigger a player's casting cooldown
 - Right click on the first rune to cast from there
 - Can form multiple spells from one starting rune, spell will split at each branch and be cast from the final sentence when the branch ends
 - Cannot cross the same rune more than once; the spell will end before trying to
 - Have a *very* high cost limit, but disappear once cast
 - Useful for big spells, like changing the time of day or summoning a powerful weapon

If you want to cast something *really* powerful, though? You might want to check out Chalk circles.

Chalk is a pretty easy tool to get your hands on, it's just made out of a couple of pieces of Calcite. But how do you actually use it?

To start, you'll probably want to know what spell you're casting in advance, and name pieces of chalk after each of the words in that spell in an Anvil. Chalk can be used an unlimited amount of times, so you only need one per unique word, but chalk will always write exactly what it's named down, and each piece can *only* be one word, so it takes a bit of planning and inventory space.

Once you have your chalk named, right click on the ground with a piece to write down that word, then write other words next to it to finish your spell. Chalk will always try to form a spell from the marking you initially use outwards, and won't loop or cross over itself, though it can branch out into multiple spells at the same time.

Chalk circles have a massive cost limit and won't impact the player's cast cooldown at all, but will disappear once you try to cast a spell with them.

Nifty things to have around, especially if you're casting spells to do things like change the time of day or summon a particularly powerful or long-lived bound weapon.

Hmm, I wonder if there's a way to make a chalk circle using a spell...?

- Lecterns
 - Cast spells from a tome automatically on cooldown
 - Cost limit is lower than a player or mob's for the same tome
 - This means it might not be able to cast a spell just because you can

Lastly, we have the champion of automation, the Casting Lectern. These are handy dandy little guys that pretty much do what you'd expect: cast a spell from a spellbook automatically.

To use one of these, just place a Spellbook inside by right clicking on the lectern with one in hand. By default, the casting limit is a bit lower for lecterns than for spells players cast directly, and the lectern will take a bit longer between casts.

Don't worry, though, the lectern will fizzle to let you know if a spell is too expensive for it or it otherwise can't meet the casting conditions. To start, a casting lectern will cast the spell placed on it as soon as it's able, off cooldown. To stop this, just provide it with a redstone signal and it'll wait to cast until the signal is removed again.

But that's not all, lecterns also have another trick up their sleeve: casting crystals. Normally, a lectern's inventory is just whatever chest it's touching, and it'll always cast as itself, facing the same direction as the binder of its book.

- Casting crystals can be used to change a spell's focus, effectively the "self" target of the spell
 - Player-focused

Casting crystals, though, let you change what the lectern's effective "self", called a Focus, is. When a crystal is changing the focus, any spells with a self subject will target that focus, and any spells cast on other things that have an effect on the caster or the caster's inventory will have that effect on the focus of the casting crystal instead. Scriptor ships with a few different casting crystals you can use for all your lectern-casting needs.

The first is the Player-focused casting crystal. This can be crafted up with a player tag, and will focus a spell on a single player. To attune it, just hold it in your hand (or have your target hold it in theirs), then use the crystal. Use it on the lectern to insert it, and while you're within 16 blocks of the lectern it'll cast its spell as if it were you! Useful for keeping yourself healed or your gear repaired when you're in your base, or to keep a specific player out, if you can convince them to attune themselves.

And to clear the attuned player, just smack it in the crafting table on its own. Easy!

- Coordinate-focused

Next up, the ever-versatile Coordinate-focused casting crystal. These are made with a coordinate tag, and can be used to focus a spell on a set of up to 4 coordinates, cycling between them after each cast. Again, this one can be used on a lectern to insert it, and will only work on blocks within 16 blocks of the lectern, casting as if the spell were cast from the block they target. They work on the inside face of a block, so if you use one on the top of the ground, it'll focus on the block above it, including any inventory on it, like chests. Great for automatically placing blocks, manipulating items, lighting fires, stuff like that!

And, just like the Player crystals, to clear out their focused coordinates, just chuck one in a crafting table.

- Nearby Animals

If you run a ranch, you might want something like the Animal-Focused Casting Crystal, though. These are a lot simpler than the previous two, focusing each cast on a random animal within a 16 block range of the lectern. This can be handy to keep your stables healed up, automatically dye your sheep a different color, or, well- [pigs on fire]

- Nearby Mobs

And last among the casting crystals, we have the Mob-Focused casting crystal. These guys are just like the animal crystals, with a very predictable twist: they work on all mobs. Good for clearing out the

zombies at the bottom of your mob farm, moving mobs out of a restricted area, or whatever you can dream up, really.

- **TODO any other new ones that are finished before the video**

- Highlight major changes

And that's it for casting! Before we move on to deeper details, I want to highlight a few of the specific changes since Scriptor 1!

- Identify Scrolls are now creative mode only

One less immediately obvious change is that Identify Scrolls have been removed from Survival mode, and are now only obtainable in Creative. In their stead, most spells have been either reworked to make their effects more obvious, moved to different casting methods to make them easier to find and break down, or attached to scraps.

- Cast cooldown is a bar now instead of just using the vanilla item cooldowns
 - Only visible when you're recovering from casting a spell

One difference you may have noticed during this video is the change to casting cooldowns! Rather than just using the vanilla item cooldown system as Scriptor 1 did, 2.0 introduces a casting cooldown bar which usually renders over your experience bar. This should hopefully make it easier to understand how the casting system works, and give you a better handle on how much spells actually cost to cast.

This bar is only visible shortly after you've cast a spell, and should keep itself out of the way if anything important is on screen.

- Most tomes, artifacts, etc. from previous versions have been removed or replaced.
 - If you discovered one of these in a previous version, you'll still be able to keep and use it

As well, pretty much every tome and a lot of the other spells in the mod have been replaced from previous versions. If you had discovered one of these in a prior version, you'll still be able to keep it around, and most of them should work more or less the same, but most of them have been replaced with spells from a handful of prolific authors, who each have their own preferences when it comes to tome colors, spell types, and naming.

- Some spell effects have changed, so spells made with them will no longer work the same

This comes with a change to how some of the spell effects work. Most of them are similar, but some which were superseded by different methods got a polish up to give them their own unique flair.

- Chain
 - Now reaches for nearby similar blocks rather than spiraling outwards

The chain descriptor, which previously spiraled outwards from the target block, now reached out to nearby blocks or entities of the same type, letting you chain between enemies and vein mine.

- Saturation > Food
 - Increased duration makes the conjured food last longer and provide more saturation
 - Increased strength makes the food fill your hunger meter more

The saturation effect, which was frankly more than a little busted, has been replaced with the conjure food effect, which provides you with temporary food that scales based on the strength and duration of the spell. This not only removes a lot of the overlap between saturation and regeneration, but opens the door to new kinds of automation using both Scriptor and other mods.

- Target descriptor grouping
 - Descriptors that modify a spell's target will only apply to the group they're in, not subsequent groups
 - This was actually changed before, as the implementation was broken at some point, but depending on your version you may not already have this change

And lastly, every descriptor which previously affected targeting for the whole spell will now only affect targeting for the section of the spell it's in. This makes spells with complex targeting conditions a bit easier to write and cast, and makes descriptors as a whole a bit easier to explain.

Okay, well that's casting out of the way, let's take a sec before moving on to cover a few other bits and bobs in the mod. Y'know, the bits I couldn't squeeze into the other categories.

- Redirect Crystal
 - Redirects projectiles that hit it in the direction it's facing
 - Useful when combined with casting lecterns, allows things like aiming vertically

And a new one in 2.0 is the Crystal of Redirection! This is a pretty basic tool with a lot of utility in base design and automation. Its function is pretty simple: when a magical projectile collides with it, it will be centered and redirected out the front of the crystal. This lets you rectify spells to let you do things like teleport around your base, sure, but it also has a lot of utility in automation!

With these, you can move projectiles around to make them more likely to hit enemies, route multiple projectiles into one place, or perform a truly forbidden art that only became possible in the most recent version: shoot projectiles vertically out of lecterns.

They aren't too tough to craft, either! All you'll need are some shards of amethyst, some blocks of glass, and a couple of wooden planks.

- Random casting bits

And to wrap up this section, some tips for casting!

- There are descriptors that can change the target of a spell, letting you do things like target other dimensions, block and entity inventories, and even yourself.
 - Might not sound super useful, but when combined with subjects like on attack and when damaged, can prove interesting

Did you know that there are descriptors to change the target of a spell? These let you do things like target other dimensions, target block inventories, or move a spell back to target yourself. This is already pretty handy, but when combined with subjects that have a fixed target, like, say, armor that casts spells when you get damaged, they can really shine.

- Some spells have specific conditions that must be met before casting, but these conditions can reduce the cost or increase the potency of the spell. Combining these effects can let you pull off pretty powerful spells

If you've played with Scriptor, you've almost certainly noticed that some spells have specific conditions that prevent them from being cast sometimes. But what's less obvious is why you might want these: descriptors like this will almost always reduce the cost or increase the potency of a spell. And when you combine these descriptors? You can pull off some pretty powerful spells without even breaking a sweat!

- Colors don't just affect the color of a spell's projectile, some subjects will use a spell's colors, too!

Okay, last tip I promise. Plenty of spells come with a descriptor that changes the spell's color. Obviously this will affect the projectile and particles the spell makes, but some actions will use a spell's color, too! Try putting down a light or a block with a color on your spell. Not to even mention what happens when you combine colors...

Alright, I think that's about everything I wanted to cover for the player's guide. That's not to say that I covered everything; the mod has a *lot* in it now, and it's hard to get to all of it. I'd suggest making a scriptonary and experimenting with spells to find out more about how everything works!

Alright, next up is the Moderator's Guide. This is mostly gonna be details for server administrators and pack devs to configure the mod, so if you just want to skip to the end where I have a few more announcements and stuff, there should be a timestamp on screen, or you can click on the timestamp I've included in the description.

- Scriptor 2.0 Moderator's Guide
 - All the options that can be configured or changed by server operators either via gamerule or config.

To start, let's go over some of the gamerules you can change. And hey, pack devs, if you want to change the default value of these gamerules, you can modify the relevant entry in the scriptor config located at **config/scriptor.cfg**. Do note that it won't have any effect on already-created worlds!

- Most casting methods have cost multipliers, as a percentage, and cost limits, which can be modified to let players cast more expensive spells or prevent them from casting without reagents
 - these are all available either in the config file before a world is generated, or in gamerules after it's been generated
 - if you generate a world and change these configs, the gamerules will not update!

Many casting methods come with a cost multiplier, which can offset the cost of this casting method, as a percent. This can limit the efficacy of casting spells via a certain method or reduce the hit to casting cooldown from casting a certain way. By cranking this value, you can even prevent players from casting spells at all without using things like reagents. By default, most all of them are 100, or 100% of the spell's cost.

These gamerules are as follows:

tome_cooldown_multiplier will affect casting through normal tomes by players.

casting_lectern_cooldown_multiplier will affect casting through tomes by lecterns.

vocal_cooldown_multiplier will affect vocal casting

scroll_cooldown_multiplier will affect casting through scrolls, and is 90% by default.

artifact_cooldown_multiplier will affect casting through artifacts, which have an item cooldown instead of affecting player cooldown. This is 200% by default

other_item_cooldown_multiplier will affect casting through inscribed items. This is 120% by default

Chalk doesn't have a player cooldown, so cannot be configured this way

Additionally, most casting methods can have their max cost tweaked or removed. This will affect how much the spell can cost before it fizzles, but bigger costs will still incur a larger cooldown on players using them.

tome_max_cost determines the max cost for tomes cast by players. This is 50 by default.

casting_lectern_max_cost determines the max cost for tomes cast by lecterns. This is 25 by default, limiting these to half the cost of spells cost by players.

chalk_max_cost determines the max cost for chalk circles. This won't affect engravings, since they don't have a max cost. By default, this is 250

scroll_max_cost determines the max cost for scrolls, and is 75 by default

`other_item_max_cost` determines the max cost of spells cast through inscribed items. This is 30 by default

`vocal_max_cost` determines the max cost for spells cast through chat or the cast command. This is -1 by default, which removes the max cost altogether.

- Other vocal casting configs

While we're on vocal casting, there are a couple of other bits you can configure.

- `vocal_hunger_threshold`

The `scriptor:vocal_hunger_threshold` gamerule determines the cost threshold after which a spell cast through chat will start to inflict hunger on the caster.

- `vocal_damage_threshold`

The `vocal_damage_threshold` gamerule is much the same, but for damage, affecting the cost after which a spell cast through chat will start to damage or kill the caster.

- `show_spells_in_chat`

The `show_spells_in_chat` gamerule affects whether spells cast vocally will show in chat. By default, this is disabled.

- Vocal casting can be disabled outright

- `cast_through_chat_enabled`

The `cast_through_chat_enabled` gamerule determines whether vocal casting is enabled at all. When disabled, vocal spells can't be cast unless the cast command is enabled.

- a cast command can be enabled, for example if vocal casting conflicts with a proximity chat mod or discord bridge

- `cast_command_enabled`

If you do disable casting through chat, the `cast_command_enabled` can be enabled to allow casting through the `/cast` command. This is useful if you have a chat mod that conflicts with scriptor, such as some proximity chat mods and discord bridges.

- scriptor also has its own basic proximity chat mode, which will prevent players from hearing each other unless they're within the specified range

- `proximity_chat`
- `proximity_chat_range`

That being said, scriptor also has its own basic proximity chat mode, which only allows players to hear each other when they're within a specified range. To enable this mode, you can toggle the `scriptor:proximity_chat` gamerule, and to change the range (in blocks) players can hear each other at you can change the `proximity_chat_range` gamerule.

- entity casting gamerules

- `entities_verbally_cast`
- `entities_verbal_proximity`

Alright, now to entity casting. Similar to the proximity chat rules, you can enable or disable entities from verbally casting with the `entities_verbally_cast` gamerule, and change the range, in blocks, at which players can hear these spells with `entities_verbal_proximity`

- other gamerules

- `can_target_player_inventories`
- `demo_do_not_generate`
- `invert_do_not_phase`

Last generally useful one here is `can_target_player_inventories`. Nowadays this is disabled by default, which prevents players from targeting each other's inventories with spells. This means that you can't, for instance, take items directly out of other people's inventories. If you want to live a bit more on the wild side, feel free to reenable it though.

And to round out the gamerules, `demo_do_not_generate` is a demo gamerule which is mostly just useful for building scriptor structures, as it prevents generate blocks from generating artifact reward boxes, engravings, and chalk circles, and `invert_do_not_phase` makes it so that only blocks in the `do_not_phase` block tag can be phased. Originally I made this for a BlanketCon display so that people wouldn't phase blocks in other people's booths, but it may also be useful in other demo settings.

- each spell component can have its cost tweaked via config file, allowing them to be rebalanced or effectively disabled if desired
 - no gamerules

Alright, gamerules done, but that's not the end of configuration! Every spell component cost in Scriptor can be tweaked, letting pack devs and server owners rebalance the mod as they see fit. To do this tweaking, just go to the `config/scriptor_cost.cfg` file in your Minecraft directory. For every action, subject, and descriptor, there will be 2 configurable entries, their type and cost. The cost is a number, and type can be one of additive, which adds to the base cost of the spell, multiplicative, which scales the base cost of the spell, and `additive_post`, which adds to the cost of the spell after all other cost calculations. This is mostly used in the base mod for descriptors that reduce the cost of a spell.

- Scriptor commands

Okay, configuration done, let's wrap up by going over Scriptor's commands. Most of scriptor's useful commands, aside from the `/cast` command, are wrapped up under `/scriptor`

- `structure` tells you the spell structure for the world and how many words were generated

`/scriptor structure` will output the spell structure for the world and how many words are in the dictionary.

- `get_action` autocompletes based on available actions, and will tell you the word for the action specified
 - `get_subject` and `get_descriptor` do the same for actions and descriptors

`get_action`, `get_subject`, and `get_descriptor` will autofill based on existing actions, subjects, and descriptors respectively, and provide you with the world's word for the component specified.

- `scriptor_debug give_dictionary` is a debug command that will give you a dictionary with every spell component already recorded

Lastly, `scriptor_debug give_dictionary` is a debug command that will give you a dictionary with every spell component already recorded in it. This won't update as new words are added, so would need to be rerun if words were added or the dictionary were regenerated for the world.

- Community mode datapack

- community mode is the same as in previous versions, where it will limit you to discovering a portion of all spells in your world, but words will be shared between your world and every other world using community mode

And before we move on, community mode! This is the same as in all previous versions, limiting you to a fifth of the spells for a world, but sharing words between your world and every other world using community mode. This is a data pack that you can add to any world using scriptor. If you add it to an existing world, it will replace the dictionary and disable a number of commands which could be used to discover words illegitimately.

- Scriptor 2.0 Meddler's Guide

- Everything that modpackers, addon developers, and datapackers might want to know

Alright, it's time to get into the weeds. Welcome to the Meddler's Guide to Scriptor, your hopefully-one-stop-shop for any and all screwery you might want to get up to. If you're here to make a modpack, addon, datapack, or anything of that sort, you'll want to check out this section.

- Overview of the online datapack generator

- more or less equivalent to making the datapack by hand, just means you don't have to keep track of as many of scriptor's idiosyncrasies

First up, the online datapack generator! It's available on its own GitHub pages, and I've included a link in the description. It's a pretty simple but useful interface for building Scriptor datapacks without having to work around Scriptor's quirks and idiosyncracies. It's not perfect, but it should be able to handle most of the things you'd want to use it for, and if it's missing a specific feature you're interested in you can feel free to open an issue or shoot me an email at blur@blur.gay about it.

- [current version is built for scriptor 2.0](#)
- [link in description](#)

- Datapacks

But, whether you're using the generator or making a datapack by hand, it's probably useful to know how they work, yeah? As such, in this section I'll be going over all the things you can do with a datapack.

First up, any json file in a scriptor datapack can have the following two keys:

The "disabled" key, when set to true, will disable the json file it's in.

The "required" key, when set to the id of a mod, will disable the json file it's in unless that mod is present.

- custom actions
 - these actions can be used to trigger specific commands when cast
 - don't scale with strength or duration, though that is planned in the future
 - can run different commands based on whether they target a position, item, or entity, though at minimum their position targeting command should be specified
 - have a special @caster target that always targets either the casting entity or a temporary entity placed at the cast position

First up are custom actions. These are spell actions that will run specific commands when cast. These have 3 different fields which allow the action to run a different command based on whether it's targeting a position, item, or entity. At the very least, though, a custom action must specify what to run when targeting a position.

Any custom action, when run, will be executed with an @s target that represents either a virtual entity at the position of the target, for cast_at_position, the targeted entity itself for cast_on_entity, or a virtual entity holding the targeted item in its main hand, for cast_on_item.

Additionally, all spells cast this way have access to a special @caster target that always either targets the entity that cast the spell or a virtual entity at the location where the spell was cast.

At the moment, custom actions don't scale with strength or duration, though the means to let them is planned for a future update.

- artifacts
 - generate rarely in loot chests, no cost limit, and don't affect player's casting cooldown
 - name can be direct or a localization key
 - otherwise only include a spell

Next up are artifacts, the rarest spellcasting medium in the game. These generate rarely in loot chests, have no technical cost limit, and don't affect player cooldown. To add one to your datapack, you only need to specify two things: the name and the spell it will cast. The name can be either a literal name or a localization key which will be substituted out during gameplay.

The spell component for an artifact is the same as for other files, so I'll go over it once here.

Each spell has a single subject, which will be the key for the subject to use, and a list of partial spells which can each contain one action and any number of descriptors. When put into the game, these will

be replaced with the generated words, restructured according to the world's spell structure, and joined with the and word as necessary.

- word generators

Next up are generators. These are what Scriptor uses to generate all the magical words for each world. For the most part, you'll only need to know about 2: Mixed Group Generators and Static Token Generators.

- mixed group
 - the default generator for most words
 - generates a word as a string of tokens pulled from non-consecutive groups

First up are mixed group generators, the style of generator that most words in Scriptor use. Mixed group generators are used to construct words from pools of character groups. These will string together entries non-consecutively from these groups with a length based on the parameters specified. To configure a mixed group generator, set the "generator" value to "mixed_groups". These generators have the following parameters:

- Parameters

The first parameters are `min_tokens` and `max_tokens`, which determined the minimum and maximum number of tokens that unconfigured words that use this generator will contain.

Next up is `max_consecutive_groups`, which determines how many times tokens from the same group can be used in a row.

Mixed group generators can also be assigned a seed, which will cause words generated with this generator to always be the same. This can be used to basically replicate community mode, though it won't enable all the same anti-cheat protections.

And lastly, but most importantly, are the groups. Each group has a weight, which determines how likely it is to be selected, and a set of tokens which will be drawn from to generate the word. A valid mixed group generator **must** contain at least 2 groups of tokens, with at least 1 token in each group.

Mixed group generators may be made default by setting the "default" field to true. If a generator is default, it will be used to generate words that aren't explicitly bound to it or another generator, using the parameters that generator was configured with.

- static token
 - cannot be a default generator
 - used to always assign a word the same token
 - each word used with it must be bound and have a number of parameters specified

Next up is the static token generator. For most use cases, you should just be able to use the default one, at `scriptor:static_generator`. This generator type will always assign a word to a specific token, forcing it to generate the same in every world.

Static generators only have one default parameter, `collision_strategy`, which is used to determine what happens when a static token would collide with an existing word. This can be `FAIL`, which will crash the game when it fails, useful for debugging data packs, `FALLBACK`, which will use the default generator if the word is already taken, or `SHORTEN`, which will keep trying to shorten the token until it doesn't collide with anything.

A static generator cannot be made a default generator, and must be bound explicitly to each word that uses it.

- community
 - if *any* words in a world use a community mode generator, the world will switch to a community dictionary and some protections will be enabled
 - this means if community mode is enabled, spells in an existing world will become invalid!

The penultimate generator is the community mode generator. This generally shouldn't be used in custom datapacks, but is available if you want. If *any* words in a world use a community mode generator, the world will switch to a community mode dictionary and enable all the associated protections, invalidating existing spells if any exist.

The community mode generator has no parameters and can be made default by setting the default field to true.

- debug
 - no parameters, just makes every word bound to it generate as if the debug pack was enabled!

Lastly is the debug generator, which you also probably don't want to use in custom datapacks since the debug pack exists. There are no parameters, any word generated with this will just be the name of the component. This can also be made default by setting the default parameter to true.

- generator bindings
 - can be used to specify how specific words will generate
 - parameters differ based on the generator they're bound to
 - static token
 - mixed group
 - debug and community mode do not have parameters

When talking about generators, I mentioned binding words to the generators. What does that mean? A generator binding is a specific set of parameters for a spell component, which affect how it generates words.

Each json file in the bindings directory lets you bind multiple words at once, and will expose a number of parameters that affect how each one binds individually.

Binding a word to a mixed group allows you to tweak only the `min_tokens` and `max_tokens` settings, letting you change the length of generated words. Stock scriptor only uses this to generate a shorter word for the "and" keyword.

Binding a word to a static token generator exposes a "token" parameter which is used directly to generate the word. Stock scriptor only uses this for an easter egg.

The community mode and debug generators don't expose any parameters for binding.

- custom colors
 - descriptors which affect the colors a spell projectile, particles, or mine will cycle through

Let's move on from bindings to custom colors. Custom colors are single colors or sets of colors which are added as descriptors. These files only have one parameter, `color`, which is a list of 6-digit rgb hex colors. If this only contains one color, spells using that color component will be that color, and if there are multiple spells using it will cycle between them in order.

By default, Scriptor uses this to add a variety of pride flag colors.

- abandoned engravings
 - engravings that spawn out in the wild
 - just specify the spell that will be cast when the golden word is used

Next up are engravings, which are abandoned engravings you might find out in the wild. These consist of only one parameter, `spell`, which is the same as spell parameters elsewhere in the datapack and will determine what spell this engraving will spawn with.

- mob inscribed items
 - specifies an item, item name, mob, equipment slot, item rarity, spawn chance, and spell
 - these items will be given to the mob specified at the rate specified, and will always drop
 - by giving a mainhand weapon with an `on_hit` spell or an equipment item with an `on_damaged` spell, you can have mobs effectively cast those spells
 - other types of spells on items will be given to the mob and be usable by players, but the mob *will not* cast the spell

And for one of the new types of spell, inscribed items. The `mob_spell_items` directory contains items inscribed with a spell and the id of the mob that that item will be given to.

These each contain a number of parameters: a name field, which can either be a literal name or a translation key to use for the item, a spell which is formatted like other spells in the datapack, the id of the item to be inscribed, the mob to be given the item, the chance that the mob will be given this item, the slot the item will be put into, and, optionally, the rarity of the item.

Items given to mobs this way *can* contain any kind of spell, but mobs will only be able to effectively use items in their mainhand or offhand with an `on_hit` effect or items in their armor slots with `on_damaged` effects. Additionally, mobs have the same spell cooldowns as players, so if they have multiple equipped items that trigger at the same time, and multiple have a spell cost greater than 0, only one will be able to effectively go off.

Scriptor contains a number of default inscribed items for Skeletons, Zombies, and Illagers.

- research notes
 - these will spawn in loot chests, and will award a specific entry in the nominomist's guide when right clicked
 - there are 2 tiers in the base mod, 1 and 2

Next up, another new feature, research notes. Each file in the **notes** directory contains 2 things: the tier to add these notes to, and the keys to the pages for these notes. Custom pages can be added through Unfocused's localized markdown pages system by placing markdown files in a namespaced assets directory under `unfocused/markdown/[lang, en_us for us english]` and added to the notes pool by putting an identifier pointing to the page in one of these notes pools.

By default, Scriptor mostly uses these to duplicate notes that can be found diagetically, to ensure that notes are accessible to players even if they don't discover the features that would ordinarily unlock them. The mod also only provides items which award notes from tier 1 and 2, so notes added to other tiers will not be available without an additional addon.

- reagents
 - just specify the item ID and the cost
 - these will be added as descriptors which reduce the spell cost by the specified amount after all other calculations and consume one of the item specified

Reagents remain in 2.0, slightly changed from before. Each entry in the reagents directory can contain 3 parameters, including a required item parameter and optional parameters for cost and power. The item parameter is the id for an item which must be consumed to cast a spell containing this reagent. If specified, the value in the cost parameter will be subtracted from the spell at the end, and the power parameter will be added to the strength of the section of the spell the reagent occurs in.

Stock Scriptor contains 3 reagents, including an amethyst and redstone reagent that modify spell cost and a gold reagent that amplifies spell strength.

- spell scraps
 - spawn in loot chests
 - these will, when used, be consumed and give the player a spell scrap with the name and description of the component rolled written on it
 - 3 tiers in the base mod, 1 through 3

Moving on, we have spell scraps, located in the scraps directory, which contain spell components that aren't attached to a spell.

These are largely the same as notes defined earlier, except that the “namespace” for the identifiers listed here is the category the spell component falls into, and the location is the name of the spell component to look up. These are specified under the “keys” parameter.

The base mod only includes items to award Scraps from tiers 1 to 3, so an addon is required to utilize any other tiers.

- spell tomes
 - the bread and butter of scriptor magicae, tomes found in loot chests will turn into these when revealed
 - 4 tiers in the base mod, 1 through 4
 - names can be localized or used directly
 - the item can be anything, but the spell will only be castable if it's a valid spellbook (other than the book of books) and only readable if it's a spellbook or a vanilla written book
 - the spell is just a spell

Our penultimate component is one of the driving forces behind spell discovery, the spell tome. Spell tomes are located under the tomes directory, and have a number of parameters available, including a name which can either be a literal name or a translation key, an author which is a literal name, a tier which determines which tier ancient tome will award this spellbook, a specific item to use for the spellbook, and a spell which is the same as spells used elsewhere in the datapack.

The item for this can be either a written book from minecraft or any of the spellbooks or obfuscated spellbooks available in Scriptor. The tier parameter can be any number, but the base mod only includes items for awarding tiers 1 through 4, so any additional tiers will require an addon.

Scriptor uses these tomes to award most spell knowledge, generally gating rarer components like cheap and reagents behind later tiers.

- engravings in villages
 - same as the other engravings, but these only spawn in villages, often near nomenclatist's houses

Last for datapacking are village engravings. This one may be a little underwhelming, but they're exactly the same as the other engravings, except these will only spawn in villages and won't spawn out in the wild.

- Scriptor debug commands

Okay, with datapacking covered, let's move on to debug commands. These are mostly intended for development, but may have utility when making modpacks or datapacks.

- give_dictionary, mentioned previously

The `scriptor_debug give_dictionary` command, mentioned previously, will give you a dictionary containing all words for a world and basic definitions for each.

- localization, which calls attention to any spell descriptions that aren't localized properly

The `scriptor_debug` localization command is another command which will point out any missing description localization for spell components. This won't work properly if executed on a server.

- `missing`, which notes which spell components don't currently appear in tomes, scraps, items, etc.

`scriptor_debug missing` is a command which will list all spell components that don't currently appear in spell items, and of those which also don't appear in scraps.

- `template_test`, which is used internally to test template pool injection and can generally be disregarded

`scriptor_debug template_test` is an internal testing command which is used to validate template pool injection and probably doesn't have much utility outside this context

- `dump`, which dumps a list of items and valid spellbooks to the console, and is used to debug spellbook accumulation and list valid IDs for tome generation. You can probably disregard this one too

And lastly, `scriptor_debug dump` will dump a list of items and valid spellbooks to the console. This is a helper used internally for spellbook validation, and likely doesn't have much utility outside of this context.

- Debug mode datapack
 - debug mode replaces every word with a generic counterpart, which can be useful for testing servers and modpacks
 - words like `action:inflame` and `descriptor:blue`

Now that we're past commands, let's go over a handful of other bits while we're still meddling.

First up is the debug mode datapack. This replaced most every word's generator with the debug generator, forcing everything to generate in a generic "category colon key" format, like `action:inflame` and `descriptor:blue`.

This is mostly just there for testing casting since I know it can get a bit tiring to discover words for a billion words while testing.

- Maven at `mvn.blur.lol`
 - Scriptor package at `com.ssblur.scriptor:common/neoforge/fabric`
 - useful functions and such exposed in the `api` package
 - Unfocused package at `com.ssblur.unfocused:common/neoforge/fabric`

And for addon developers, it's also worth knowing where the mod sources can be imported. There's a maven up at `mvn,blur.lol` with `scriptor` and `unfocused` packages exposed in both the snapshot and releases channels under `com.ssblur.scriptor-common/neoforge/fabric` and `com.ssblur.unfocused-common/neoforge/fabric`. Scriptor specifically exposes a number of useful classes and helper methods through the `com.ssblur.scriptor.api` package.

And that's it for the meddler's guide! For those of you who jumped forwards from earlier in the video, welcome to the part where I walk about all the other interesting stuff I wanted to talk about!

- Scriptor Magicae Mana Mode, an already-available mod that changes Scriptor's behavior for those interested
 - Casting cooldowns for players are replaced with a mana pool you can cast from that regenerates over time.

Did you know that mana mode wasn't just an April Fools' joke? Scriptor Magicae: Mana Mode is an addon for Scriptor that replaces the default casting cooldown bar with a more traditional mana resource that is depleted when casting a spell and regenerates over time. You can download it right now from CurseForge and Modrinth.

- ~~Scribeblock: Magica Arte, an in-development modpack where you have to rebuild a flooded world from zero by learning the words that bind it. (TODO: maybe hold off until done?)~~
- Huge thanks
 - Everyone who helped with testing the betas
 - Ender & Team Voided, who helped with some dev and consulted on design decisions
 - Translators who worked on 1.0
 - The-Forgiven-Zero
 - KingAlexey212
 - gelatinoustesseract
 - Kasualix
 - Rivmun
 - thekillergoose
 - Players and Viewers!

I also want to give a huge huge huge thanks to everyone who helped with Scriptor 2.0, including all the beta testers in the Discord, Ender and Team Voided for helping with debugging and design during development, and all the translators who worked on Scriptor 1.0, including The-Forgiven-Zero, KingAlexey212, gelatinoustesseract, Kasualix, Rivmun, and thekillergoose.

And lastly, but certainly not leastly, thank *you*! Like you you, the person watching this video or playing with Scriptor. I've enjoyed working on this mod and it's nice to know that people enjoy something I made so thoroughly.

Oh, and bonus, if you're interested in articles about the game design decisions, real-world religious inspirations, and more about Scriptor Magicae, I should have one out by the time this video comes out and I'll likely post more. There's a card on screen and a link in the description.

... also while we're here let's look at the to-do list from the last video.

IDK that Scriptor is feature-complete but it's certainly a lot more featureful.

Yep, there are more words, of all sorts. Oh, and new casting modes like scrolls, magical enchantments, and artifacts that are at least *shaped* like a staff.

Scriptor's datapack support has never been stronger!

And last up, could do with more mod support. Sure, there's Alchimiae support and stuff, but what about other people's? Y'all'll have to let me know what mods you'd like to see Scriptor compat with, if I can think of anything that works well for 'em I might add that stuff in.

Oh, and if I forgot anything or you have any questions, leave a comment, join my discord, or shoot me an email at blur@blur.gay.

Anyways, whew, this whole thing was like a year plus in the making. I hope y'all enjoy the new Scriptor update, I think I'm gonna enjoy some time off.

Until next time, take it easy!