Configuration

The configuration file is located in plugins/ViaVersion/config.yml. It uses the [YAML](#) Syntax. The config is split into sections, you can see these by looking at the comments prefixed with #.

A quick reminder on YAML:

- Boolean means it can either be `true` or `false`, (true being on and false being off).
- String means that it is a sentence or words surrounded by `""`, for example `message: "Hello world!"`
- Integer means that it is a number, for example: `limit: 10`
- Integer List means that it can be empty, `[]` or have as many values as desired, `[10, 20, 30]`
- Map means that it has a string key followed by a colon then the value, eg. `{hello:1, world:2}`

Tip

A common mistake when using .yml (YAML) is using the tab key, ensure that you use spaces instead.

A recommended editor for Windows is [Notepad++](#).

Global Options

These options affect all versions, even when it isn't being handled by ViaVersion.

`checkforupdates` - default: true (Boolean)

This option indicates whether we should check for updates when the server boots and when an operator logs in, this can be a huge benefit to ensure that bugs are fixed as soon as new versions are available. We use [spiget.org](#) to handle checking for updates, so ensure that it is allowed through your firewall if you wish to use this feature.

`send-supported-versions` - default: false (Boolean)

When a client sends a ping request to your server, for example viewing the Minecraft server list should we send extra information telling them what versions are supported? We suggest only using this if you use some third party software to handle what servers a client can connect to.

`block-protocols` - default: [] (Integer List)

This allows you to block certain versions from logging into your server, every Minecraft version has a special number assigned to it called a Protocol Version. You can view a list of them [here](#). For example, if you wanted to block 1.10 Minecraft you can see from that list that the number is 210. You would then change your config to, `block-protocols: [210]`, you can then add as many protocols as required separated by a comma if you wish to block more.

`block-disconnect-msg` - default: "You are using an unsupported Minecraft version!" (String)

When the condition above for blocking a certain version is met, what should the client be kicked with? This message will appear on their disconnect screen when they are kicked for using a blocked version. You can use [Minecraft Colour](#) codes here.

`reload-disconnect-message` - default: "Server reload, please rejoin!" (String)

If you use ProtocolLib we are unable to reload the plugin without crashing the server with players connected, to protect against this we kick players. Here you can select what message is sent to them on reload, furthermore you should really use a Plugin Manager instead of /reload due to memory leaks. You can use [Minecraft Colour](#) codes here.

Global Packet Limiter Options

When building ViaVersion we came across a lot of issues with exploits where people could send lots of packets and crash a server (Just as a general Minecraft fault). To combat this we included a simple packet limiting system, which should work out of the box.

Tip

If you don't want packet limiting, set `max-pps` to -1 and `tracking-period` to -1.

Maximum based limiting

A simple way to protect your server is to set the maximum packets per second, this way if a malicious client sends too many attempting to overload your server they are disconnected.

`max-pps` - default: 800 (Integer)

This is how many packets the client can send in 1 second, using the value of -1 will disable this. From research the most intensive task in Minecraft is fishing while being in a boat where we've observed averages of 90-100 pps. Though after lagging from something like a resource pack download this can exceed this to about 400.

`max-pps-kick-msg` - default: "You are sending too many packets!" (String)

When a player sends more than the limit above what should the kick message be. You can use Minecraft Colour codes here, as well as %pps which will be replaced with their current packets per second.

Monitoring packets over a period of time

Although maximum packets per second is nice, in cases where it's valid to send lots of packets (For example: Having latency after logging in and sending lots to catch up in a second). Monitoring them over a period allows you to punish players who go over a number of packets per second several times over the period.

`tracking-period` - default: 6 (Integer)

This is measured in seconds and represents how long one tracking period is, every x seconds it will check how many seconds the client has been over the `tracking-warning-pps`. You can set this to -1 to disable limiting based on periods of time.

`tracking-warning-pps` - default: 120 (Integer)

How many packets per second should be counted as a warning, you should set this as a value you would not expect an average client to obtain.

`tracking-max-warnings` - default: 4 (Integer)

Over the tracking period how many warnings is the client allowed to have before being kicked, usually you want it so that on average they are sending more packets than normal. The maximum value for this is `tracking-period`.

For example if you have `tracking-period` of 6, `tracking-warning-pps` of 120 and `tracking-max-warnings` of 4: If a client sends 120 packets or more for 4 seconds out of 6 seconds they will be kicked.

`tracking-max-pps-kick-msg` - default: "You are sending too many packets!" (String)

When the player gets more warnings over the time period than allowed and kicks them, what shall the message be. You can use Minecraft Colour codes here, as well as %pps which will be replaced with their current packets per second.

Bungee Options

BungeeCord is very different to our other platforms we support as it's a proxy. This means we have to do some fancy stuff to work out what version the internal servers are to connect your users to. On large networks, some of our most basic options may be inefficient for your setup and you may want to revert to doing things using our manual methods.

By their name these options are only available on BungeeCord.

`bungee-ping-interval` - default: 60 (Integer)

To be able to get the versions of servers on your network we send them a ping list (like the Minecraft server list in your client), this option indicates how long to check the protocol. **You can set this to -1 to disable pinging.** (You will still be able to use /viaversion probe though!)

`bungee-ping-save` - default: true (Boolean)

If this option is enabled and `bungee-ping-interval` is enabled, when the protocol version is not the same as what's in the config it will add/replace it. You could use this to initially populate your config with values then disable the `bungee-ping-interval`.

bungee-servers - default: (Integer Map)

Example

```
bungee-servers:
  default: 47
```

The default will be filled with the lowest version your BungeeCord can handle. You can view a list of them here. The key is the BungeeCord server name from your config.yml and the integer is the protocol version from the previous link.

When a client connects to a server ViaVersion will gather the protocol in the following order:

1. If the server is in `bungee-servers`, use the protocol version provided there.
2. If pinging is enabled, use the last pinged protocol version for the server.
3. Use the `default` key under `bungee-servers`.
4. Fallback to the lowest version the server can handle. (Should never happen)

Multiple Versions Options

These options affect multiple versions.

hologram-patch - default: false (Boolean)

In 1.9 and 1.11, certain height values for holograms were modified. If you used Armour Stands manually on your server this may not concern you but if you use a plugin such as Holographic Displays then your armour stands may appear higher than they should be to 1.9+ but also may be lower on 1.11+. To fix this you can enable this option.

hologram-y - default: -0.96 (Integer)

When the `hologram-patch` above is true, how much should be offset the hologram to 1.9 and above clients. Our experiments show that -0.96 works the best. For 1.11 this is inverted.

piston-animation-patch - default: false (Boolean)

Should we disable piston animation for 1.11/1.11.1 clients? In some cases when firing lots of pistons it crashes them.

chat-nbt-fix - default: true (Boolean)

In 1.12 and above nbt data displayed in chat had a slight format change with removing the labels of indexed arrays, this option will strip that data.

quick-move-action-fix - default: true (Boolean)

Due to changes in Minecraft in 1.12 shift double click does not work properly for some items, this patch enables sending extra info to fix this (only works on 1.8-1.11.2 bukkit based servers).

team-colour-fix - default: true (Boolean)

Due to changes in Minecraft in 1.13 nametag colour is based on team colour and not prefix, this fix changes it so it's the last colour from prefix.

suppress-1_13-conversion-errors - default: false (Boolean)

Due to changes in Minecraft in 1.13 some conversion errors may happen with unusual block ids / item ids / biome ids, this will hide warnings we print to the console.

disable-1_13-auto-complete - default: false (Boolean)

1.13 has a new tag complete which may trigger kicked for spam on Spigot, you can disable it completely with this option.

`1_13-tab-complete-delay` - default: 0 (Integer)

Introduces a tick delay (20 ticks per second) to tab completion, if another request happens within the ticks it will be cancelled. This can be used to allow tab complete but not frequently, eg. 100 ticks being one completion every 5s.

`serverside-blockconnections` - default: false (Boolean)

In 1.13 Mojang changed block connections to be handled by the server, because of this we need to calculate the connections required for blocks (this includes across chunks), this feature will have performance impact as additional calculations may be required when handling fences, panes, doors etc.

`block-connection-method` - default: packet (String, Either *world* or *packet*)

Due to the way block connections work, we offer a packet-level version and a block level version which will use internals in your server to calculate the blocks to connect. It's worth noting that only packet is available on proxy software and the overhead of the packet-level block connections is significantly bigger (more memory and more processing). However, the world mode may be unstable on some server software and cause crashes depending on the server software (though no fatal world losses have been reported).

`reduce-blockstorage-memory` - default: false (Boolean)

Due to the overhead of storing blocks we offer the option to reduce this by not storing non-important blocks. When this option is enabled fences, glass panes etc. won't connect to solid blocks.

`flowerstem-when-block-above` - default: false (Boolean)

This is a block connections option where flower stems are used for decorative purposes, enabling them will stop us from connecting them.

`change-1_14-hitbox` - default: false (Boolean)

This changes the hitbox so that 1.14 players can sneak under blocks, this may allow players to be missed by projectiles.

`truncate-1_14-books` - default: false (Boolean)

The page limit is now greater in 1.14 so this will limit pages at 50 instead of 100 (As some anti-exploit software may kick players).

`fix-low-snow-collision` - default: false (Boolean)

1.13 doesn't have collision for 1 layer of snow, this will send 2 layers of snow to have a similar effect.

1.9 & 1.10 clients on 1.8 servers options

This options only apply to you if you use a 1.8 server, these enable consistent visuals / gameplays across future versions where features have changed.

`prevent-collision` - default: true (Boolean)

In 1.9, players can now push each other. To prevent this when scoreboard teams are sent ViaVersion can set the collision to not happen so that the gameplay is balanced. (Plugins like ColoredTags use teams)

`auto-team` - default: true (Boolean)

Most servers don't use scoreboard plugins, in this case we send our own team to the player to prevent them colliding with other players. If you use a plugin like ColoredTags you should turn this off. Furthermore if you are having issues with Bungee and teams, consider turning it off too.

`suppress-metadata-errors` - default: false (Boolean)

Due to changes in 1.9 and above to how entity data is sent some plugins send data that we can't find the 1.9 equivalent for (Usually means they're doing something wrong!). When this happens we'll tell you all the info in your config and you will need to debug and work out which plugin is causing this and contact the author, (these are related to the NMS class called Datawatcher).

Alternatively, you can just suppress metadata errors but this means mobs may look different on 1.9 to 1.8, but most of the time this is fine.

`shield-blocking` - default: true (Boolean) - Note: On BungeeCord the client will not be able to see them self blocking.

In 1.9, the blocking animation when you right click a sword was removed. It's not possible to simulate meaning 1.9 would not be able to see 1.8 blocking, as an alternative you can visually see these as shields. These do not change any game play and just allow 1.9 to see clients blocking and perform the blocking action with a sword themselves.

Disabling this will mean that 1.9 clients will not be able to see players blocking and will not be able to block properly.

`simulate-pt` - default: true (Boolean)

In 1.9, the player ticking is no longer triggered by the client sending packets. Therefore if we don't simulate sending them packets they won't be able to eat, use bows properly, drink potions. If you have a really simple server you can disable this but it's suggested that you do not.

`nms-player-ticking` - default: true (Boolean) - *This feature is only available on Spigot / Sponge. (On BungeeCord it will use packet ticking)*

To simulate player ticking we use NMS as an alternative to sending packets as sending packets can cause issues with anti-cheats. If you are experiencing TPS issues with ViaVersion it's suggested you try with this to false as it may reduce the effect on TPS. (Though it will still be there just not labelled as ViaVersion as it's internal server ticking)

`bossbar-patch` - default: true (Boolean)

In 1.9 bossbars are handled differently and sent via packets rather than using a mob. Should we automatically send the right packets to enable bossbars to work? You can disable this if you're having issues with too many boss bars appearing.

`bossbar-anti-flicker` - default: false (Boolean)

Due to the change in bossbars some plugins constantly update the health on the boss bar which may cause flickering, enabling this option will fix the health. However this will prevent the health from moving down and is not recommended unless you're having flickering issues.

`use-new-effect-indicator` - default: true (Boolean)

In 1.9, a new effect indicator was introduced in the top left of the screen. Enabling this option will allow players to see their active potion effects in the top left, this may be an advantage to 1.9+ players and you may wish to disable if you find this so.

`use-new-deathmessages` - default: false (Boolean)

In 1.9, death messages were added to the respawn screen. Enabling this option will allow players who died to see their death message on the respawn screen.

`item-cache` - default: true (Boolean) - *This feature is only available on Spigot / Sponge. (On BungeeCord it will still work but be less efficient causing the server to send 1 extra packet)*

In 1.9, packet changes to how items are used means that they aren't sent to the server. Item cache allows ViaVersion to save what items players are holding, this allows an efficient way of telling the server what items the player is using where it has been removed. Disabling this is not suggested as the alternative is looking up the item as the packet arrives which can cause latency and possibly lead to crashing a server.

`anti-xray-patch` - default: true (Boolean) - *This feature is only available on Spigot.*

Due to the way ViaVersion works to convert chunks to the new chunks for 1.9, anti-xray may not function properly. This option only works with Spigot anti-xray, and will be forcefully disabled if you do not use Spigot (So don't worry). If you don't use anti-xray with spigot you can disable this function for a slight performance boost.

`replace-pistons` - default: false (Boolean)

This option also affects 1.9/1.9.1 servers. In 1.10.1, Mojang introduced a way to ensure special blocks had data sent. This also introduced a crashing issue which meant when block 36 (extending piston special block) was sent it would crash the client. As a fail safe you can enable this option and we'll send the replacement id specified below instead. (This issue would only effect you if you use the special block 36)

`replacement-piston-id` - default: 0 (Integer)

When the previous option is enabled what block id should be sent instead, by default it is air 0 as block 36 is very similar to air.

`force-json-transform` - default: false (Boolean)

When an issue happens when sending json, should we forcefully still send it instead of throwing an error? This may cause incorrect json to be sent on rare occasions and disabling it will show the responsible errors.

`minimize-cooldown` - default: true (Boolean)

Attempt to reduce the visual cooldown on item hitting in newer versions than 1.8.

`minimize-cooldown` - default: true (Boolean)

1.9 introduced a visual cooldown indicator, this option tells the client to lower the speed of this (this is just a graphical change)

`left-handed-handling` - default: true (Boolean)

This enables left hand to be used on 1.8 servers for newer clients.

`change-1_9-hitbox` - default: false (Boolean)

This changes the 1.9 player's hitboxes so that there are no knockback bugs that occur when players' are sneaking.