

# Clearlag 1.12

This script removes all items from world every 30 minutes. Only works in 1.12.

```
// Create item whitelist filter that won't be deleted with clearlag
var whitelist = Ingredient.matchAny([
    'minecraft:diamond', // Adds diamond to whitelist
    'minecraft:gold_ingot',
    '@tinkersconstruct', // Adds all items from tinkersconstruct to whitelist
    'minecraft:emerald'
])

// Create variable for last clearlag result
var lastClearLagResult = Utils.newList()
// Create variable for total number of items
var lastTotalClearLagResult = Utils.newCountingMap()

// Create new function that clears lag
var clearLag = server => {
    // Get a list of all entities on server with filter that only returns items
    var itemList = server.getEntities('@e[type=item]')
    // Create new local map for item counters
    var lastResult = Utils.newCountingMap()
    // Clear old result lists
    lastClearLagResult.clear()
    lastTotalClearLagResult.clear()
    // Iterate over each entity in itemList and add item counters
    itemList.forEach(entity => {
        if (!whitelist.test(entity.item)) {
            // Get the name of item
            var key = entity.item.name
            // Add to entity count
            lastResult.add(key, 1)
            // Add to total item count
            lastTotalClearLagResult.add(key, entity.item.count)
            // Kill the item entity
            entity.kill()
        }
    })
}
```

```

}))

// Update and sort last result list
lastClearLagResult.addAll(lastResult.entries)
lastClearLagResult.sort(null)

// Tell everyone how many items will be removed
server.tell([
    Text.lightPurple('[ClearLag]'),
    ' Removed ',
    lastTotalClearLagResult.totalCount,
    ' items. ',
    Text.yellow('Click here').click('command:/clearlagresults'),
    ' for results.'
])
}

// Listen for server load event
events.listen('server.load', event => {
    // Log message in console
    event.server.tell([ Text.lightPurple('[ClearLag]'), ' Timer started, clearing lag in 30 minutes!' ])
    // Schedule new task in 30 minutes
    event.server.schedule(MINUTE * 30, event.server, callback => {
        // Tell everyone on server that items will be removed
        callback.data.tell([ Text.lightPurple('[ClearLag]'), ' Removing all items on ground in one minute!' ])
        // Schedule a subtask that will clear items in one minute
        callback.data.schedule(MINUTE, callback.data, callback2 => {
            clearLag(callback2.data)
        })
        // Re-schedule this task for another 30 minutes (endless loop)
        callback.reschedule()
    })
})

// Doesnt work in 1.16+!
// Register commands
events.listen('command.registry', event => {
    // Register new OP command /clearlag, that instantly runs clearlag
    event
        .create('clearlag')
        .op()

```

```

.execute(function (sender, args) {
    clearLag(sender.server)
})
.add()

// Register new non-OP command /clearlagresults, that displays stats of all removed items from previous
/clearlag
event
.create('clearlagresults')
.execute((sender, args) => {
    sender.tell([ Text.lightPurple('[ClearLag]'), ' Last clearlag results:' ])

    lastClearLagResult.forEach(entry => {
        var total = lastTotalClearLagResult.get(entry.key)

        if (entry.value == total) {
            sender.tell([ Text.gold(entry.key), ': ', Text.red(entry.value) ])
        } else {
            sender.tell([ Text.gold(entry.key), ': ', Text.red(entry.value), ' entities, ', Text.red(total), ' total' ])
        }
    })
})
.add()
})

```

Revision #4

Created 28 June 2020 17:25:02 by Lat

Updated 14 October 2022 14:22:41 by Lat