New:

<u>Global Player/Forge Events (Backported):</u>
Access by right clicking the scripter in the air and clicking on either the 'Players' or 'Forge' tab.

Scripting is in the style of the 1.12 CNPC version, with scripts for events being written as functions invoking them. Up to 100 scripting tabs are available to write in, and an exhaustive list of function names to invoke events is available to the left of the scripting GUI. Scripts written in these tabs will run for events globablly on the client/server, so it's best to make them as efficient as possible to minimize lag. Unlike in 1.12, function tick() for player events does run on every tick. To make it run on a higher frequency, ten ticks for example, surround your code with:

```
if(world.getTime()%10){
...
}
```

<u>NPC API (Backported):</u>
The keyword "API" is available in all scripting APIs now. Calls an instance of the NpcAPI class, which has plenty of useful functions. Consult the 1.12 API reference for a list of all functions available to you.

<u>ScriptEntityParticle:</u>

Spawns custom particles from any directory on entities. Add as many images you want to a resource pack, then use them as particles!

Create a particle with a predefined directory through: API.createEntityParticle(directory). then modify its attributes with the following functions:

- setDirectory(string directory) - Sets a new directory for the particle
- setHEXColor(int color) - Sets the color of a particle from a hex integer (0xFFFFFF by default)
- setAmount(int amount) - Sets the amount of particles to spawn
- setMaxAge(int age) - Sets the number of ticks the particle will be alive for
- setPosition(double x, double y, double z) - Sets the position of the particle relative to the entity it's spawning on
- setPosition(double x, double y, double z, float gravity) - Sets the motion of the particle in the x y and z directions. The particle accelerates downwards when gravity > 0, and accelerates upwards when gravity < 0.
- setScale(float scale1, float scale2, float scaleRate, int scaleRateStart)
  scale1 : The starting scale of the particle
  scale2 : The maximum/minimum scale of the particle if it grows/shrinks
  scaleRate : The rate at which the particle grows/shrinks. Negative value shrinks the particle, a positive value expands it.
  scaleRateStart : When the scale rate begins occuring, in ticks.
- setAlpha(float alpha1, float alpha2, float alphaRate, int alphaRateStart)
  alpha1: The starting transparency of the particle (0 to 1)
  alpha2: The maximum/minimum transparency of the particle if it grows/shrinks
  alphaRate: The rate at which the particle fades or appears. Negative value makes the particle fade, positive value makes the particle appear.
  alphaRateStart: When the alpha rate begins occuring, in ticks.

Then, call particle.spawnOnEntity(entity) to spawn it on an entity. Each of the fields in the functions above have separate getters and setters as well. Sends a packet from the server to the client on each call of spawnOnEntity(entity).

<u>ScriptBlock (Backported):</u>
Allows access to blocks in the world as CNPC scripting objects. This is **not** the addition of custom scripted blocks, just a much more versatile return value when getting a block from a world with “world.getBlock()”.

<u>ScriptBlockPos (Backported):</u>
Positions in the world can be represented as CNPC scripting objects now! Has some useful functions for calculating distance between two blocks, adding/subtracting positions, etc.

<u>ScriptContainer (Backported):</u>
Modifies data for blocks that contain items like chests and crates. Retrieved from ScriptBlock by calling block.getContainer() where 'block' is a ScriptBlock.

<u>ScriptNbt (Backported):</u>
NBT Tag support for scripting. Tags can be created, deleted, or modified now, with some classes like ScriptItemStack and ScriptPlayer having functions such as getNBT().

<u>Timers for players and NPCs (Backported):</u>

A new “Timer” hook is added for NPCs, and a “timer” event hook is available for players, which runs scripts when a timer on a player or NPC reaches 0. Timers have IDs, lengths, can repeat, and are accessed by the getTimers() function call for either ScriptPlayer or ScriptNpc.

To start a timer, call player.getTimers().start(int id, int length, bool repeat). If the player already has a timer with the same ID running, the console will throw an error message. To start a timer regardless of whether another timer with the same ID is running or not, call player.getTimers().forceStart(id,length,repeat). If not, the timer will run for “length” ticks and repeat if “repeat” is true.

To stop a timer with a given id, call player.getTimers().stop(id). To check if a player has a timer, call player.getTimers().has(id), returning a true value if a timer is running, and false otherwise. To reset a timer, call player.getTimers().reset(id). To stop and erase all timers, call player.getTimers().clear().

For all of these examples to work on an NPC, simply call them with the “npc” keyword instead of “player”.

Changes/Additions to existing features:

ScriptItemStack:
-Modified giveItem() method to give the player item(s) preserving NBT attributes
-Added getNbt() and getItemNbt()

ScriptWorld:
-Added getDimensionID()
-Added createEntityParticle(directory) for creating instances of ScriptEntityParticle

ScriptEntity:
-Added spawnParticle(scriptEntityParticle)
-Added getWidth(), getYOffset(), and getHeight()
-Added new knockback(x,y,z,direction) function with parameters for knockback power in the x y and z axes
-Added setImmune(ticks), which makes the entity immune to all damage until “ticks” amount of ticks. If an entity is being hit and is currently in its invulnerable state, calling entity.setImmune(0) will allow it to be hit again.
-Added storeAsClone(tab,name)
-Added getNbt(), getAllNbt(), and setNbt(ScriptNbt)

ScriptLivingBase:
-Added hurt(damage), hurt(damage, ScriptEntity) and hurt(damage, ScriptDamageSource). Hurts the entity by “damage” amount of damage, with the source coming from either an entity or another damage source.

ScriptNpc:
-Added setVisibleTo(player,visible). When “visible” is false, the NPC will not be visible to this player. When true, it’s visibility is overridden by the NPC’s visibility type. On a server, setting setVisibleTo(player,false) for Steve, and not doing so for another player, Alex, means Alex will be able to see the NPC, but Steve won’t.
-Added isVisibleTo(player). True if the NPC has had its visibility toggled for a specific player, i.e., setVisibleTo(player,false) was called for that player. False otherwise.
-getTimers(), used to manipulate an NPC’s timers.

ScriptPlayer:
-Added setPosition(x,y,z,dimensionID), teleports a player to the position in the dimension matching the given dimension id.
-Added getDimension(), gets the dimensionID of the player.
-Added updatePlayerInventory(), which updates inventory changes on the client-side
-Added getTimers(), used to manipulate an player’s timers.
-Added checkGUIOpen(), which checks whether the player has a GUI open. Sends a packet from the server to client, updates PlayerData. Takes two function calls on two separate ticks to return the correct value.
-setRotaion(yaw,pitch) now correctly changes the player’s rotation
-removeItem() functions now properly update the player’s inventory