@ ? d: -> =>

Lonelog: Resource Tracking Add-on

Inventory, Supplies, and Wealth Notation

Version 1.2.0



Table of contents

Resource Tracking Add-on **1**
Overview 1
1. The Inventory Tag 4
2. Abstract Supply Tracking 10
3. Wealth and Currency 14
4. Resource Events 17
5. The Resource Status Block 20
Cross-Add-on Interactions 22
System Adaptations 25
Best Practices 28
Quick Reference 31
FAQ 34
Credits & License 36

Resource Tracking Add-on

Overview

Some games don't care what's in your pack. Others treat every torch, every ration, every last arrow as a decision that matters. This add-on is for the second kind.

Core Lonelog already handles light resource tracking — you can put `Gear:Flashlight,Notebook` inside a `[PC:]` tag and call it a day. That works fine when resources are flavor. But when your game makes resource management a *mechanic* — when running out of torches in a dungeon means something, when the Usage Die ticking down creates real tension, when you need to know exactly how many silver pieces you have left — you need more structure.

That's what this add-on provides: a dedicated notation layer for tracking resources as they flow in and out of your game.

If your system handwaves supplies or resolves resource scarcity with a single oracle question, you don't need this add-on. Core Lonelog's `[PC:]` tag with a `Gear:` field is more than enough.

What This Add-on Adds

- `[Inv:Item|×N|props]` — Track individual items with a count and properties. Example: `[Inv:Torch|×3]`
- Supply notation — Abstract resource levels inside `[PC:]`. Example: `[PC:Kael|Supply d8]`
- `[Wealth:Owner|Currency N]` — Currency and trade tracking. Example: `[Wealth:Kael|Gold 45|Silver 12]`
- `[RESOURCES]` / `[/RESOURCES]` block — Snapshot of current resources at session boundaries. Example: See §5

No new core symbols. This add-on introduces no additions to `@`, `?`, `d:`, `->`, or `=>`.

Design Principles

Track what the game tracks. If your system uses Usage Dice, use the Usage Die notation. If it counts individual arrows, count individual arrows. If it handwaves supplies, don't invent tracking that isn't there.

Separate what you carry from who you are. The `[PC:]` tag describes your character — HP, stress, conditions, abstract resources that feel like stats. The `[Inv:]` tag describes your stuff — concrete, countable things that come and go. This separation keeps both tags readable as your campaign grows.

Record changes at the point of fiction. Don't maintain a separate inventory spreadsheet you update silently. Show resources changing *when they change*, inline with the actions and consequences that cause it. Your log should tell the story of your resources, not just their current state.

Block tags follow the core convention. Structural blocks use `[BLOCK]` / `[/BLOCK]` bracket syntax — identical to `[COMBAT]` / `[/COMBAT]` in the Combat Add-on. For analog notebooks, use `--- BLOCK ---` / `--- END BLOCK ---` dashed separators instead.

1. The Inventory Tag

The `[Inv:]` tag tracks a specific, concrete item or resource. It's the thing you can pick up, use, drop, trade, or lose.

Format:

```
[Inv:Item|×count|properties]
```

Fields:

- `Item` — the item name, unique enough to search for in a log
- `×count` — how many you have, written `×N` like any count in core Lonelog (§4.1); omit it for unique or singular items
- `properties` — freeform; use for weight, condition, magical status, ammunition type, or anything else your game cares about

Examples:

```
[Inv:Torch|×3]
[Inv:Rope|×1|50ft]
[Inv:Healing Potion|×2|magical|restores
d8 HP]
[Inv:Rations|×5|days]
[Inv:Arrow|×12]
```

Unique or singular items (no quantity needed):

```
[Inv:Skeleton Key|unique]
[Inv:Father's Compass|quest]
[Inv:Map to the Ruins]
```

1.1 Gaining Items

When you acquire something, introduce it with a full `[Inv:]` tag, just as you introduce an NPC with `[N:]`:

Example — minimal:

```
=> I find supplies! [Inv:Torch|×4]
[Inv:Rope|×1|50ft] [Inv:Gold|×25]
```

Example — expanded:

```
@ Search the chest
d: Investigation d6=5 vs TN 4 -> Success
=> Inside: torches, rope, some coin.
[Inv:Torch|×4] [Inv:Rope|×1|50ft]
[Inv:Gold|×25]

@ Loot the fallen guard
=> [Inv:Short Sword|×1|rusty]
[Inv:Rations|×2|days]

tbl: d100=67 -> "A strange amulet"
=> [Inv:Bone Amulet|×1|unknown|magical?]
```

Multiple items at once — list them; no special syntax needed:

```
=> The merchant's pack contains:
   [Inv:Torch|×6] [Inv:Rope|×2|50ft each]
[Inv:Rations|×10|days]
```

```
   [Inv:Flint & Steel|×1] [Inv:Waterskin|
×1|full]
```

1.2 Consuming and Losing Items

Two approaches, same result — choose what fits your flow.

Approach A: Inline with consequences

Depletion happens inside the `=>` line, as part of the fiction:

```
=> The tunnel is pitch dark. I light a
torch. [Inv:Torch|×3→2]
=> I bandage the wound with my last
healing potion. [Inv:Healing Potion|×0]
=> Arrow flies true. [Inv:Arrow|×12→11]
```

Approach B: Standalone update

Depletion gets its own line, separate from narrative:

```
=> The tunnel is pitch dark. I light a
torch.
[Inv:Torch|×-1]

=> I bandage the wound.
[Inv:Healing Potion|×-1]
```

Which to use? Approach A is more compact and keeps resource flow visible within the fiction. Approach B is clearer when multiple things change at once, or when you want to visually separate narrative from bookkeeping. Many players mix both depending on context.

Quantity changes follow core's rules for counts:

- `[Inv:Torch|×-1]` — used one
- `[Inv:Torch|×+2]` — found two more
- `[Inv:Torch|×1]` — sets the count: one left
- `[Inv:Torch|×3>1]` — sets it, showing the old value too (`→` works as well)
- `[Inv:Torch|×0]` — none left

Keep `depleted` for things that run out without being counted, like a lantern's oil or a flashlight's battery.

1.3 Item State Changes

Items don't just appear and disappear — they break, get repaired, become cursed, run out of charges. Use properties to track this:

- `[Inv:Short Sword|×1|rusty]` — initial state
- `[Inv:Short Sword|rusty>repaired]` — one property replaced by another
- `[Inv:Short Sword|enchanted]` — a property added
- `[Inv:Short Sword|-rusty|sharpened]` — one removed, one added

Condition shorthand:

```
[Inv:Lantern|×1|broken]
[Inv:Shield|×1|cracked]
[Inv:Rations|×3|spoiled]
[Inv:Wand|×1|charges 2/5]
```

These are core Lonelog's tag update rules (§4.1): properties accumulate, `-` removes one, `old>new` replaces one.

Charges and uses are always written `charges N/M`, whether the game calls them charges, uses or doses. Spend one with `charges-1`: `[Inv:Wand|charges-1]`.

1.4 Reference Tags for Items

Just like NPCs, use # for items already established:

The first mention gives the full details:

```
[Inv:Bone Amulet|×1|unknown|magical?]

... later in the log ...

@ Examine the amulet more closely
d: Arcana d6=6 vs TN 5 -> Success
=> [#Inv:Bone Amulet] — it's a ward
against undead!
[Inv:Bone Amulet|unknown>Ward of the
Dead|identified]
```

1.5 Grouped and Bulk Items

Some games track bundles or containers rather than individual items. Count the bundle, or list a container's contents as a property:

```
[Inv:Arrow|×20]
[Inv:Adventuring Kit|×1|contains: rope,
pitons, torch×2, rations×3]
[Inv:Quiver|×1|contains: arrow×15]
```

For slot-based inventory systems:

```
[Inv:Slot 1|Short Sword]
[Inv:Slot 2|Shield|cracked]
```

```
[Inv:Slot 3|Torch×3]
[Inv:Slot 4|empty]
```

1.6 Shared and Stored Gear

Not everything you own is on your back. Two categories cover it:

- `owner:` — who the item belongs to, when it isn't the character whose log this is: `[Inv:Rope|×1|50ft|owner:Party]`. In group play, Partylog's `[Party:Crew|…]` tag does the same job for shared resources.
- `where:` — where it is, when that isn't on your person: `[Inv:Spare Torches|×6|where:base camp]`.

Both are categories, so moving an item replaces its location (core §4.1):

```
=> I stash the spare torches at the cave
mouth.
[Inv:Spare Torches|where:cave mouth]
=> Back for them on the way out.
[Inv:Spare Torches|where:carried]
```

2. Abstract Supply Tracking

Not every game counts individual items. Many solo systems use abstract mechanics to represent "how well-supplied are you?"—Usage Dice (Black Hack, Macchiato Monsters), Supply tracks (Ironsworn), or simple qualitative levels. These feel more like character stats than inventory, so they live in the `[PC:]` tag alongside HP, Stress, and other abstract attributes.

2.1 Usage Dice

The Usage Die is a staple of OSR-adjacent solo play. You have a die representing supply level—roll it when you use a resource. On a 1–2, the die steps down. When it steps below d4, the resource is gone.

Format:

```
[PC:Kael|Supply d8]
[PC:Kael|Torchlight d6]
[PC:Kael|Ammo d10]
```

Example — minimal:

```
d: Supply d8=2 -> Step down!
=> [PC:Kael|Supply d8→d6]
```

Example — expanded:

```
@ Make camp, use supplies
d: Supply d8=2 -> Step down!
=> Supplies dwindling. [PC:Kael|Supply
d8→d6]
```

```
@ Fire another volley
d: Ammo d6=1 -> Step down!
=> Running low on arrows. [PC:Kael|Ammo
d6→d4]

@ Light another torch
d: Torchlight d4=2 -> Depleted!
=> Last torch sputters out. [PC:Kael|
Torchlight depleted]
```

The step-down chain: `d12 → d10 → d8 → d6 → d4 → depleted`

2.2 Supply Tracks

Some systems use numbered tracks for supply—similar to progress tracks but counting down as resources are spent.

Format:

```
[PC:Kael|Supply 5/5]
```

Example — minimal:

```
=> I make camp and eat. [PC:Kael|Supply
5→4]
```

Example — expanded:

```
```

```
=> I make camp and eat. [PC:Kael|Supply
5→4]
=> I forage successfully. [PC:Kael|Supply
4→5]
=> Desperate — I eat the last of it.
[PC:Kael|Supply 1→0|starving]
```

This is functionally the same as core Lonelog's `[Timer:]` used for resources. The difference is semantic: a Timer is a narrative pressure device, while a Supply track inside `[PC:]` is a character stat. Use whichever framing matches your game.

2.3 Qualitative Levels

For games that don't use numbers at all — or when you want to track a resource loosely:

```
[PC:Kael|supplies:abundant]
[PC:Kael|supplies:adequate]
[PC:Kael|supplies:low]
[PC:Kael|supplies:critical]
[PC:Kael|supplies:depleted]
```

Updating: because `supplies:` is a category, writing it again replaces the old level (core §4.1):

```
=> After three days in the wastes, food
is scarce.
[PC:Kael|supplies:low]
```

You can define your own levels. These aren't hard-coded — use whatever vocabulary your game provides or that feels right.

3. Wealth and Currency

Money behaves differently from gear. You don't usually track each coin as an "item" — you track totals, and those totals change through spending, earning, looting, and trading.

3.1 The Wealth Tag

For games with concrete currency, use the `[Wealth:]` tag. Its name is the owner — a character, the party, a ship's purse — so each purse stays one element as its amounts change, and several characters can each have one:

Format:

```
[Wealth:Kael|Gold 45|Silver 12|Copper 30]
[Wealth:Kael|Credits 1500]
[Wealth:Kael|Caps 87]
```

Example — minimal:

```
=> Good price! [Wealth:Kael|Gold+15]
=> [Wealth:Kael|Gold-8] [Inv:Rations|×5|
days] [Inv:Rope|×1|50ft]
```

Example — expanded:

```
@ Sell the jeweled dagger to the merchant
d: Persuasion d6=5 vs TN 4 -> Success
=> Good price! [Wealth:Kael|Gold+15]

@ Buy rations and a new rope
```

```
=> [Wealth:Kael|Gold-8] [Inv:Rations|×5|
days] [Inv:Rope|×1|50ft]
```

Full state vs. delta updates:

- `[Wealth:Kael|Gold 45]` — explicit total
- `[Wealth:Kael|Gold+15]` — earned 15
- `[Wealth:Kael|Gold-8]` — spent 8
- `[Wealth:Kael|Gold 45→52]` — was 45, now 52

3.2 Abstract Wealth

Some games don't count coins — they use wealth levels or resource rolls:

- `[PC:Kael|wealth:comfortable]` — a qualitative level
- `[PC:Kael|Wealth d8]` — a Usage Die for wealth
- `[PC:Kael|Resources 3/5]` — a track

These live in `[PC:]` rather than `[Wealth:]`, because abstract wealth feels more like a stat than a ledger.

3.3 Trade and Barter

For transactions, record both sides:

```
@ Trade the amulet for passage
=> I hand over the amulet. [Inv:Bone
Amulet|×0]
=> The captain agrees to take me north.
[Thread:Passage to Northport|Open]

@ Barter rations for information
=> [Inv:Rations|×-2] The fisherman tells
me about the sea caves.
```

No special syntax needed — the existing consequence notation handles trades naturally. The key is recording *what left* and *what arrived*.

4. Resource Events

Resources interact with the rest of your game. These patterns show how resources connect to actions, oracle questions, and ongoing pressures.

4.1 Resource Checks

When the game asks “do you have enough?”:

```
@ Cross the frozen river
? Do I have enough rope?
-> Yes (checking: [#Inv:Rope] 50ft —
river is ~30ft wide)
=> I anchor the rope and cross safely.
[Inv:Rope|×1|50ft|frayed]

@ Camp in the wilderness
d: Supply d8=1 -> Step down!
=> Food's running low. [PC:Kael|Supply
d8→d6]
? Do I find water nearby?
-> No, but...
=> A dry streambed — if I follow it,
maybe tomorrow.
[Timer:Dehydration 2]
```

4.2 Scarcity as Oracle Modifier

Some systems modify oracle likelihood based on resource state. Note it:

```
? Can I find more arrows in the ruins?
(Unlikely)
(note: remote area)
d: d6=4 -> Yes, but...
=> I find a quiver with only 3 usable
arrows. [Inv:Arrow|×+3]

? Is there food in this abandoned camp?
(Very Unlikely)
(note: the camp is old)
d: d6=1 -> No, and...
=> The food is rotted and the smell
attracts something. [Clock:Predator 1/4]
```

4.3 Resources in Combat

When the Combat Add-on is in use, resource consumption integrates naturally:

```
[COMBAT]
Rd1
@ Fire bow at Orc 1
d: Ranged d6=5 vs TN 4 -> Success
=> Arrow hits! [F:Orc 1|HP-3] [Inv:Arrow|
×-1]

Rd2
@ Throw last flask of oil
d: Ranged d6=3 vs TN 4 -> Fail
=> Missed! The oil splatters uselessly.
```

```
[Inv:Oil Flask|depleted]
[/COMBAT]
```

4.4 Resources in Dungeon Crawling

When the Dungeon Crawling Add-on is in use, resource depletion tracks alongside room exploration:

```
[R:4|status:active|Fungal cavern|exits:E
R5, W R2]

@ Navigate the cavern carefully
d: Survival d6=4 vs TN 4 -> Success
=> I find a path through. Torch is
getting low.
[Inv:Torch|×2→1] [Timer:Torchlight 3]

@ Search for useful fungi
tbl: d6=5 -> "Glowing moss — edible"
=> [Inv:Edible Moss|×3|restores 1 HP
each]
[R:4|status:cleared, looted]
```

5. The Resource Status Block

For long sessions or campaigns where resources matter, a snapshot at session boundaries helps you pick up where you left off.

Format:

```
[RESOURCES]
[PC:Name|stats]
[Wealth:Owner|currencies]
[Inv:items...]
[/RESOURCES]
```

Example — digital:

```
[RESOURCES]
[PC:Kael|HP 12/15|Supply d6|Stress 2]
[Wealth:Kael|Gold 52|Silver 8]
[Inv:Short Sword|×1|sharpened]
[Inv:Shield|×1|cracked]
[Inv:Torch|×2]
[Inv:Rations|×3|days]
[Inv:Rope|×1|50ft|frayed]
[Inv:Healing Potion|×1|restores d8 HP]
[Inv:Arrow|×9]
[Inv:Bone Amulet|×1|Ward of the Dead]
[/RESOURCES]
```

Example — analog:

```

```

```
--- RESOURCES ---
PC:  Kael | HP 12/15 | Supply d6 | Stress
2
Wealth: Gold 52 | Silver 8
Inv: Short Sword (sharpened), Shield
(cracked)
     Torch ×2, Rations ×3 days
     Rope 50ft (frayed), Healing Potion
×1
     Arrow ×9, Bone Amulet (Ward of the
Dead)
--- END RESOURCES ---
```

When to use it:

- At the **end of a session** to freeze state
- At the **start of next session** as a recap
- **Before a major expedition** (dungeon dive, long journey, dangerous mission)
- After **significant loot or loss** — a natural checkpoint

When to skip it:

- Resources aren't mechanically important in your game
- The session is short and changes are minimal
- You can reconstruct state easily from the log

Cross-Add-on Interactions

Situation	Add-on(s) Used	Key Tags/Blocks
Dungeon exploration with supply pressure	Dungeon Crawling + Resource Tracking	`[R:]`, `[Inv:]`, `[Timer:]`
Combat with ammunition tracking	Combat + Resource Tracking	`[F:]`, `[COMBAT]`, `[Inv:]`
Trading in a settlement	Resource Tracking only	`[Inv:]`, `[Wealth:]`, `[N:]`
Full dungeon crawl with fights	All three add-ons	`[R:]`, `[COMBAT]`, `[Inv:]`, `[Wealth:]`

Combined example — all three add-ons:

```
[RESOURCES]
[PC:Kael|HP 12/15|Supply d6]
[Inv:Torch|×2] [Inv:Arrow|×9]
[Inv:Healing Potion|×1]
[Wealth:Kael|Gold 52]
[/RESOURCES]

S5 *Entering the crypt*
[R:1|status:active|Entry hall, bones
everywhere|exits:N R2, E R3]

@ Light a torch and move in
[Inv:Torch|×-1] [Timer:Torchlight 6]
```

```
? Are there enemies?
d: d6=6 -> Yes, and...
=> Skeletons rise from the bone piles!

[COMBAT]
[F:Skeletons|×3|HP 3 each|pos:Close]

Rd1
@ Fire at the nearest skeleton
d: Ranged d6=5 vs TN 4 -> Success
=> Arrow shatters its skull!
[F:Skeletons|×2] [Inv:Arrow|×-1]

@(Skeleton) Charge at me
d: Attack d6=3 vs TN 4 -> Fail
=> Clumsy swing, I dodge.

Rd2
@ Draw sword, press the attack
d: Melee d6=6 vs TN 4 -> Success
=> Cut clean through! [F:Skeletons|×1]

@ Finish the last one
d: Melee d6=4 vs TN 4 -> Success
=> The crypt goes quiet. [F:Skeletons|×0]
[/COMBAT]

@ Search the room
tbl: d20=14 -> "A locked chest"
? Can I pick the lock?
d: Lockpicking d6=5 vs TN 5 -> Success
```

```
=> Inside: [Inv:Gold Bracelet|×1|
valuable] [Wealth:Kael|Gold+10]
[R:1|status:cleared, looted]
[Timer:Torchlight-1]

@ Move north toward R2
[R:2|status:active|Ritual chamber|exits:S
R1, D R3]
```

System Adaptations

The Black Hack / Usage Dice Systems

Usage Dice are the primary resource mechanic. Track them in `[PC:]`:

```
[PC:Varn|Torches d8|Rations d6|Ammo d10|
HP 14/18]

@ Set up camp
d: Rations d6=2 -> Step down!
=> [PC:Varn|Rations d6→d4]
(note: one more step-down and they're
gone)
```

Ironsworn / Supply Track

Supply is a single stat from 0–5:

```
[PC:Kael|Supply 4/5|Health 4/5|Spirit
3/5|Momentum 6/10]

@ Sojourn at the settlement
d: Sojourn d6+2=7 vs 3,8 -> Weak Hit
=> I resupply but time passes. [PC:Kael|
Supply+2|Momentum-1]
```

OSR / Encumbrance Systems

Slot-based inventory is common in OSR games:

```
[Inv:Slot 1|Sword]
[Inv:Slot 2|Shield]
[Inv:Slot 3|Torch×3]
[Inv:Slot 4|Rations×4]
[Inv:Slot 5|Rope 50ft]
[Inv:Slot 6-10|empty]
(house: 10 slots max, STR-based
encumbrance)

@ Pick up the golden idol
=> [Inv:Slot 6|Golden Idol|heavy|occupies
2 slots]
[Inv:Slot 7|occupied]
(note: at 7/10 slots — movement penalty
at 8+)
```

Fate / Narrative-First Systems

Resources are aspects or stress tracks, not inventories:

```
[PC:Sable|Aspect:Well-Provisioned]
[PC:Sable|Aspect:Well-Provisioned→Running
Low]
[PC:Sable|Aspect:Running Low|invoked
against me]
```

Or simply note it in prose and skip `[Inv:]` entirely. If your game doesn't mechanize resources, neither should your notation.

Survival Horror / Scarcity Games

When every bullet counts:

```
[Inv:9mm Round|×7]
[Inv:First Aid Kit|×1|charges 2/3]
[Inv:Flashlight|×1|battery d4]

@ Fire at the creature
d: Firearms d6=4 vs TN 4 -> Success
=> Hit! [Inv:9mm Round|×-1]

@ Patch up the wound
=> [Inv:First Aid Kit|charges-1]
[PC:Casey|HP+3]

@ Check the flashlight
d: Battery d4=1 -> Step down!
=> Flickering badly. [Inv:Flashlight|×1|
battery d4→depleted]
=> Darkness. [Clock:Panic 1/4]
```

Best Practices

Do: Record resource changes at the point of fiction

```
✔ @ Light a torch to see
  => The cavern reveals a narrow passage.
[Inv:Torch|×-1]
```

Don't: Silently update resources outside the log. If you subtract three arrows, write it down, even briefly:

```
✔ [Inv:Arrow|×-3]
```

Do: Use the Resource Status Block at session boundaries

```
✔ [RESOURCES]
  [PC:Kael|HP 12/15|Supply d6]
  [Inv:Torch|×2] [Inv:Arrow|×9]
  [/RESOURCES]
```

Don't: Let bookkeeping overwhelm the fiction. Track what matters this beat, and batch the rest in a status block.

```
✗ => I fight the orc.
  [Inv:Arrow|×-1] [Inv:Arrow|×11]
[PC:Kael|HP-2] [PC:Kael|HP 13]
  [F:Orc|HP-4] [Clock:Alert+1]
[Timer:Torch-1]
```

```
  (note: this is a spreadsheet, not a
story)

✔ => Arrow finds its mark — the orc
staggers.
  [Inv:Arrow|×-1] [F:Orc|HP-4]
```

Do: Match your notation to your system's resource model

```
✔ Usage Die game:    [PC:Kael|Supply d8]
✔ Counting game:     [Inv:Arrow|×12]
✔ Narrative game:    [PC:Sable|
Aspect:Well-Provisioned]
```

Don't: Track resources the game doesn't care about (unless your game literally tracks bootlaces)

```
✗ [Inv:Bootlaces|×2|leather]

✔ Only track what creates meaningful
decisions or tension.
```

Do: Separate concrete items (`[Inv:]`) from abstract stats (`[PC:]`)

```
✔ [Inv:Healing Potion|×2]
✔ [PC:Kael|Supply d6]
```

Don't: Put an inventory list inside a character stat tag

```
✗ [PC:Kael|HP 12|Torch 3|Arrow 9|Rations
5]

✔ [PC:Kael|HP 12|Supply d6]
  [Inv:Torch|×3] [Inv:Arrow|×9]
[Inv:Rations|×5]
```

Quick Reference

New Tags

- `[Inv:Item|×N|props]` — Concrete inventory item. Example: `[Inv:Torch|×3]`
- `[#Inv:Item]` — Reference previously established item. Example: `[#Inv:Bone Amulet]`
- `[Inv:Item|×+N]` / `[Inv:Item|×-N]` — Quantity change (gain/lose). Example: `[Inv:Arrow|×-1]`
- `[Inv:Item|×N>M]` — Explicit quantity transition. Example: `[Inv:Torch|×3>2]`
- `[Inv:Item|depleted]` — Item fully consumed or gone. Example: `[Inv:Oil Flask|depleted]`
- `[Wealth:Owner|Currency N]` — Concrete currency total. Example: `[Wealth:Kael|Gold 45]`
- `[Wealth:Owner|Currency+N]` — Currency earned. Example: `[Wealth:Kael|Gold+15]`
- `[Wealth:Owner|Currency-N]` — Currency spent. Example: `[Wealth:Kael|Gold-8]`

Abstract Resources in `[PC:]`

- `Supply d8` — Usage Die resource. Example: `[PC:Kael|Supply d8]`
- `Supply d8→d6` — Usage Die step-down. Example: `[PC:Kael|Supply d8→d6]`
- `Supply 4/5` — Supply track. Example: `[PC:Kael|Supply 4/5]`
- `supplies:low` — Qualitative level. Example: `[PC:Kael|supplies:low]`
- `Wealth d8` — Abstract wealth as Usage Die. Example: `[PC:Kael|Wealth d8]`

- `Aspect:Running Low` — Narrative resource state. Example: `[PC:Sable|Aspect:Running Low]`

Structural Blocks

Block	Opens When	Closes When
`[RESOURCES]` … `[/RESOURCES]`	Session start/end, before major expeditions	After all current `[Inv:]` and `[PC:]` tags are listed

Usage Die Step-Down Chain

```
d12 → d10 → d8 → d6 → d4 → depleted
```

Complete Example

```
[RESOURCES]
[PC:Kael|HP 10/15|Supply d6]
[Wealth:Kael|Gold 52]
[Inv:Torch|×2] [Inv:Arrow|×9]
[Inv:Healing Potion|×1]
[/RESOURCES]

@ Light a torch
[Inv:Torch|×-1] [Timer:Torchlight 6]

@ Fire at the orc
d: Ranged d6=5 vs TN 4 -> Success
=> [F:Orc|HP-4] [Inv:Arrow|×-1]
```

```
=> I bandage up after the fight.
[Inv:Healing Potion|×-1] [PC:Kael|HP+5|HP
15]
```

FAQ

Q: When should I use `[Inv:]` vs. just putting gear in `[PC:]`? A: If resources are mechanically important and change often, use `[Inv:]`. If they're background flavor or you're only tracking one or two things, `[PC:Name|Gear:Sword,Shield]` is fine. There's no wrong answer — use what keeps your log clear.

Q: Do I need the `[Wealth:]` tag, or can I put money in `[Inv:]`? A: Either works. `[Wealth:]` is cleaner for multi-currency systems (gold/silver/copper) and for money that flows frequently. `[Inv:Gold|×45]` works fine for simpler games or if you're already using `[Inv:]` for everything.

Q: Should I write a Resource Status Block every session? A: Only if it helps you. If resources are central to your game (dungeon crawls, survival horror), a block at session start and end saves you from scrolling back. If resources rarely change or aren't mechanically important, skip it.

Q: What if I'm playing a game with no resource mechanics at all? A: Then you probably don't need this add-on. Core Lonelog's `[PC:]` tag with a `Gear:` field is more than enough. This add-on exists for games where resources create tension and decisions.

Q: Can I combine `[Inv:]` with the Dungeon Crawling Add-on's `[R:]` tags? A: Absolutely — they're designed to work together. Tag loot in the room where you find it: `[R:3|status:+looted]` `[Inv:Gold Ring|×1|valuable]`. The `[R:]` tag tracks what happened to the room; the `[Inv:]` tag tracks what happened to the loot.

Q: How do I handle containers, bags of holding, saddlebags? A: Either nest descriptively — `[Inv:Bag of Holding|contains: Wand, Scrolls×3, Gold 100]` — or track the container and contents separately. For slot-based systems, just mark which slots are "in the bag." Don't over-engineer it.

Q: The `×` symbol is hard to type. Can I use `x` instead? A: Yes. `[Inv:Arrow|×12]` and `[Inv:Arrow|x12]` are both fine. Readability over formality.

Q: How does this work in an analog notebook? A: The Resource Status Block works well in analog — write it in a box or across the top of the page. For inline changes, use shorthand in the margin: `Torch ×-1`, `Arrow ×9→8`. The `+/-` notation is fast enough that it doesn't slow down handwriting.

Credits & License

© 2025 Roberto Bisceglie

This add-on extends Lonelog by Roberto Bisceglie.

Version History:

- v 1.2.0: Charges are always `charges N/M`; new §1.6 on shared and stored gear (`owner:`, `where:`). Aligned with Lonelog v1.7: `[Wealth:]` is named after its owner (`[Wealth:Kael|Gold 45]`); named `[PC:]` tags, standard `d:` line for Ironsworn rolls. Quantities are `×N` counts (`[Inv:Torch|×3]`, `[Inv:Torch|×-1]`); qualitative levels use a `supplies:` category; likelihood on the question; explanations moved out of code blocks.
- v 1.1.0: Block tags unified with `[BLOCK]`/`[/BLOCK]` convention; design principles updated
- v 1.0.0: Rewritten as a compliant add-on (previously "Resource Tracking Module")

This work is licensed under the **Creative Commons Attribution-ShareAlike 4.0 International License**.

You are free to share and adapt this material, provided you give appropriate credit and distribute adaptations under the same license. Session logs and play records created using this add-on's notation are your own work and are not subject to this license.