PoWR v1.1 — Reference & Mining Specification

1. Overview

PoWR (Proof-of-Work Reference) is a lightweight proof-of-work scheme embedded directly into a human-readable bibliographic reference.

A valid PoWR reference consists of:

1. A **reference body** (structured plain text)
2. A **12-hex-digit PoWR code**
3. A **64-hex-digit SHA-256 hash**
4. A **difficulty condition** derived from expiry parameters

A reference is **validly mined** if the SHA-256 hash of the reference body (including the PoWR code, excluding the final hash) is **numerically less than a calculated maximum hash value**.

2. Reference Structure (Exact Format)

A complete PoWR reference **MUST** be formatted as follows:

```
[REFERENCE BODY | PPPPPPPPPPPP]
[HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH]
```

Where:

- `PPPPPPPPPPPP` = 12 hexadecimal characters (PoWR code)
- `HHHH...` = 64 hexadecimal characters (SHA-256 hash)
- Square brackets `[ ]` are the **only reserved delimiters**
- All other characters are treated as literal content
- No characters in-between main reference and hash section.

3. Reference Body Requirements

The **reference body** is free-form text but **MUST** include the following required fields, written exactly as key–value pairs:

Required Fields

Field	Requirement
`Cited:`	Date the reference becomes valid
`Expires in:`	Lifetime in seconds
`	`

Example (minimal valid structure)

```
[GENESIS Author (Year). Title. Publisher.
Cited: 2026-02-02.
Expires in: 259200s.
Claim: "Example".
Scope: test.
| 000000000000]
```

Notes:

- Field order is **not enforced**, but spelling **is**
- Dates **must** be `YYYY-MM-DD`
- Expiry **must** be an integer number of seconds followed by `s`
- The PoWR code placeholder **must exist** before mining

4. PoWR Code

- Exactly **12 hexadecimal characters**
- Case-insensitive
- Treated as part of the reference body
- Modified during mining

Example:

```
| 0000000001e5
```

5. Hashing Rules

5.1 What Is Hashed

The SHA-256 hash **MUST** be computed over:

```
REFERENCE BODY INCLUDING:
- all text
- the PoWR code

EXCLUDING:
- the final [hash] block
```

In other words:

```
hash = SHA256( "[reference body | PoWR code]" )
```

The result:

- Rendered as lowercase hexadecimal
- Exactly 64 characters
- Appended as the final bracketed block

6. Expiry Validity Rules

A reference is **temporally valid** if:

```
Cited date ≤ current time ≤ (Cited date + Expires in seconds)
```

If outside this window, the reference is invalid **regardless of mining correctness**.

7. Difficulty & Maximum Hash Calculation

7.1 Constants

These constants are fixed by the PoWR v1.1 standard:

```
2^256 = total SHA-256 hash space
baseExpiry = 10800 seconds (3 hours)
desiredMiningTime = 1 second
```

⚠️ **Important:**
`hashesPerSecond` is **NOT a requirement**.
It is a **difficulty calibration factor** chosen by the verifier.

7.2 Expected Attempts

Let:

```
E = expirySeconds
H = hashesPerSecond (chosen constant)
```
Then:

```
expectedAttempts = (H × desiredMiningTime × E) / baseExpiry
```

7.3 Maximum Acceptable Hash

```
maxHash = floor( 2^256 / expectedAttempts )
```
This defines the mining difficulty.

8. Proof-of-Work Condition

A reference is **validly mined** if:

```
numeric(SHA256(reference body)) < maxHash
```
Where:

- The hash is interpreted as an **unsigned 256-bit integer**
- Comparison is purely numeric

9. Mining Process (Normative)

To mine a reference:

1. Insert a 12-hex-digit PoWR code
2. Compute SHA-256 of the reference body
3. Check if hash < maxHash
4. If not, modify the PoWR code and repeat
5. On success, append the final hash

No other fields may be altered during mining.

10. Genesis & Chaining (Optional but Defined)

- A **GENESIS** reference:

    - Has no previous hash
    - Must be the first reference in a chain
- A continuation reference:
    - Must embed the previous reference's hash verbatim
    - Verification **must fail** if hashes do not chain

11. Verified Example (Valid until 2026/12/03)

```
[GENESIS John Doe (2026-02-03). Piece of work. Journal
Article. John's university. Locator: example.com. Version: 1.
Language: en. Viewed: 2026-02-03. Cited: 2026-02-03. Expires
in: 25920000s. Claim: "n/a". Scope: n/a. | 0000001a0c78]
[0000000a76403dbf24d61300036bbe67ad43eb51d5a3d3d02139ff6cd15f
6d06]
```