

DRCE - Following an engineering claim from source to a corrected release

Selected evidence companion · curated September 16, 2026

The Development Record and Candidate Evidence system (DRCE) connects an evidence-grounded engineering history to readable and machine-readable releases. Its engineering problem is preserving meaning through investigation, correction, compilation and delivery. A convincing narrative is insufficient if its source supports a different mechanism, its test is mistaken for a test result, or its machine representation drops a qualification.

Justin created and led the system's design, integration and qualification with extensive Codex implementation and investigation. This exhibit follows one corrected historical claim from a completed real update. It shows DRCE's contribution: a precise accepted account, explicit source relationships, preservation of unaffected history and a release whose verification can be inspected. Protocol Runner coordinated the investigation and review; it did not decide historical truth or release acceptance.

The selected update reviewed 595 commits and added 40 accounts of significant engineering developments, containing 190 claims and 196 attribution routes. Three accounts required bounded repair. The example below is one of those three: process identity and timestamp normalization in the machine-control tooling.

1. A claim specific enough to challenge

The practical issue was process ownership. A recorded process ID can later belong to a different process. The implementation began saving a freshly checked creation time with the ID, then corrected comparison of equivalent timestamps across Windows PowerShell and PowerShell Core. That is a narrower, more useful account than broadly crediting the change with runtime reliability.

The accepted episode's claim is reproduced below. Its long evidence-route field and two additional boundary fields are omitted here; the next section shows the relevant source excerpts. All displayed lines are verbatim. Source: `Development_Record/Episodes/E194.md`, lines 35-43, under DRCE's system documentation at snapshot `ef8546aaa`.

```
### E194-001
```

- Proposition: Managed startup persisted process-creation identity and failed closed on ambiguous ownership, then normalized equivalent PowerShell time forms without weakening PID-reuse defense.
- Class: `git_established`
- Class detail: Exact permitted Git objects establish the bounded source or declared-authority fact.
- Status: established

The same claim carries these qualifications, also verbatim:

- Limit: Source presence establishes only the stated implementation or authority fact; it does not by itself establish execution, effect, readiness or general support.
- Limit: The mechanism does not guarantee that every managed service is healthy or eliminate all workstation interruption modes.

Within a DRCE release, stable identifiers let an agent follow the claim without relying on prose search. The evidence class says what establishes it. The limitations travel with the proposition rather than living in an unrelated disclaimer.

2. Inspect the source mechanism

The claim's retained attribution route identifies the startup change at commit 232f9790 and the representation correction at ce75d558, including exact Git object identities, source files, relevant sections and what each supports. This complete function is from the latter change, on August 27, 2026, in `scripts/tools/local_runtime_common.ps1`, lines 517-534:

```
function ConvertTo-UtcIdentityDateTime {
    param(
        [object]$Value
    )

    if ($Value -is [DateTimeOffset]) {
        return ([DateTimeOffset]$Value).UtcDateTime
    }
    if ($Value -is [DateTime]) {
        return ([DateTime]$Value).ToUniversalTime()
    }

    return [DateTimeOffset]::Parse(
        [string]$Value,
        [System.Globalization.CultureInfo]::InvariantCulture,
        [System.Globalization.DateTimeStyles]::RoundtripKind
    ).UtcDateTime
}
```

The distinction is visible in the code: equivalent time representations become comparable; process-creation identity is not discarded. The associated test obtains fresh process identity, compares it with saved state, then asks both PowerShell hosts for runtime status. Its cross-shell assertion loop is reproduced verbatim from `systems/workstation_control/tests/test_ops_tool_handshakes.py`, lines 255-272, at the same commit:

```
for shell_name, status_runner in (
    ("powershell", _run_powershell),
    ("pwsh", _run_pwsh),
):
    status = status_runner(
        "scripts/tools/status_local_runtime.ps1",
        "-RuntimeId",
        runtime_id,
        "-CatalogPath",
        str(catalog_path),
    )
    assert status.returncode == 0, (
        f"shell={shell_name}\nSTDOUT:\n{status.stdout}\nSTDERR:\n{status.stderr}"
    )
    payload = json.loads(status.stdout)
    runtime = payload["runtimes"][0]
    assert runtime["processIdentity"]["identityMatches"] is True
    assert runtime["staleState"] is False
```

This is an inspectable test oracle: a representation mismatch or stale identity would fail the assertions. Its presence establishes an authored test, not a retained execution result. The historical account preserves that distinction.

3. Correct the account without damaging unrelated history

The corrected episode explicitly identifies timestamp representation as the problem, excluding repository-root or child-environment normalization. This matters because the wrong abstraction would give a reviewer an inaccurate account of what was diagnosed and built.

The completed update's operational record, DRCE Operations/Maintenance.md lines 201-205 at snapshot ef8546aaa, says:

Bounded repairs to E190/E193/E194 preserved the other 37 episode objects and all claim/evidence mappings.

That is a retained completion statement, not a newly rerun comparison. It binds the repair to a corrected integrated draft and a fresh independent challenge with zero open findings. All 169 predecessor source files were also preserved byte-for-byte. Preservation here means this bounded update kept unrelated accepted work intact; it does not make accepted history immune to later evidence-based correction.

The retained review workflow required a fresh reviewer, separated the reviewer from the coordinating writer, and prohibited access to prior or sibling review answers. The reviewer could assess the supplied candidate and admitted evidence but could not repair or integrate it. These are specific measures against reviewing the coordinator's rationale as if it were independent evidence.

4. Preserve the claim through human and machine output

DRCE compiles one selected typed record set into human documents, structured records, evidence traces and search material. This excerpt from packages/compiler/src/compiler.ts within the DRCE system, lines 263-268 at snapshot ef8546aaa, shows that the readable form includes evidence routes, limitations and boundaries:

```
...claim.evidence_routes.map((route) =>
  ` - Evidence: ${route.evidence_id} | ${route.role}${route.note === undefined ? "" : ` |
${route.note}`}`
),
...claim.limitations.map((limit) => ` - Limit: ${limit}`),
...Object.entries(claim.boundaries ?? {}).map(([key, value]) => ` - Boundary ${key}: ${value}`),
...(claim.unresolved_questions ?? []).map((question) => ` - Unresolved: ${question}`),
```

The compiler serializes that selected record set into JSONL and constructs claim-to-evidence links from its evidence routes. These are representations of one account, not independent corroboration. The architectural decision prevents separately maintained human and agent versions from silently drifting. Hashes and deterministic builds establish content identity and reproducibility; source inspection and review establish whether the account is supported.

5. Inspect the accepted result

The completed cycle accepted internal release drce-internal-20260902.1. Below is a field selection from its committed machine-readable receipt, reformatted for readability. Omitted fields concern release bindings, accounting and private custody; the displayed values are unchanged.

```
{
  "accepted": true,
  "reviewed_commit_count": 595,
  "admitted_event_count": 541,
  "outcome": "mixed",
  "hold_reasons": [],
  "verification": {
    "assembled_views": true,
    "challenge_findings_open": 0,
    "compiler": true,
    "contracts": true,
    "docs": true,
    "independent_challenge": true,
    "manifests": true,
    "repository": true
  }
}
```

The accompanying record reports two deterministic builds, archive-member readback, exact source/metadata checks and canonical acceptance of that same prepared candidate. Cloud custody used one archive upload followed by exact-version full-byte readback; immediate reconciliation used zero further uploads and verified the same local recovery receipt. Those are retained execution results from this completed cycle, distinct from the source and test exhibits above.

This is evidence of a working internal history-to-release pipeline with correction and reproducibility, not merely a proposed documentation format. Real career-material publication remains separate. Later catch-ups do not alter the identity or earned result of this release; the latest catch-up has already been accepted by Justin, while its remaining archive-execution work is a separate technical step.

The historical source and accepted claim are pinned to their original revisions. Compiler and completion-record inspection use committed snapshot ef8546aaa7a79c006f75a68da027389912b1f2d1, September 16, 2026. These selected excerpts are curated for inspection; they do not grant access to the private repository, raw investigation conversations or operational custody material.