Universal Manifest v0.5 — Registry Profiles

Draft registration candidates

v0.5.0. Published as tag v0.5.0 of UniversalManifest/universal-manifest, with the same wire files as release candidate 1 (v0.5.0-rc.1). It is a mechanism release, not a completed implementation or conformance qualification. Immutable versions in UniversalManifest/universal-manifest are normative; main is the working version. Earlier versions were reviewed in the OMA3 Technical Working Group, and publication asserts no OMA3 ratification.

8 November 2026

This document: /spec/v0.5/registry-profiles/

Document set: Universal Manifest v0.5.0 · What changed in v0.5.0

Artifacts: JSON Schema · JSON-LD context

History: https://universalmanifest.net/spec/ (version index and changelog)

Author: Grig Bilham — inventor, architect, and author of the Universal Manifest.

Copyright © 2026 SumSet Tech LLC and the Universal Manifest contributors. This specification is available under the Apache License, Version 2.0. It is free and remains free. Earlier published revisions retain their original licence notices.

---

v0.5.0. Published as tag `v0.5.0` of `UniversalManifest/universal-manifest`, with the same wire files as release candidate 1 (`v0.5.0-rc.1`). Wire version `0.5`; namespace `https://universalmanifest.net/ns/v0.5`; reading routes `/spec/v0.5/`. It is a mechanism release, not a completed implementation or conformance qualification. Feature-specific PREVIEW and conditional requirements remain in force.

This is the **draft standalone profile registry** required by EXT-OPT §O6. It prepares the list of Universal Manifest profiles and subsidiary value registries that share §O6's registration process. Its draft tables do not independently establish registration. Once published under §O6, this registry is **versioned independently of the specification** and is updated between specification releases.

Status: This document is PREVIEW — it is the first issue of the registry, the local draft that §O6 describes, with the maintainer decisions below adopted for v0.5. Identifiers marked `provisional` are not registered and confer no conformance standing.

Maintainer decisions for v0.5.

1. **The post-quantum suite is not registered in v0.5.** "Signature Profile B" means JCS + ES256 in EXT-OPT §O15; EXT-T2 §T2.3.2 uses the descriptive name **JCS + ML-DSA-65**. Its identifier, verification and carrier parameters still need §O6 review and conformance integration, which is later-version work. No additional profile letter is assigned.
2. **`provisional` is adopted as a listing status.** §O6 defines three statuses for registered profiles (`active`, `deprecated`, `sunset`) and requires at least one conformance fixture before registration. Most profiles named across the document set are design-stage and have no fixture. This registry lists them as **`provisional`**: named and reserved against collision, not registered, and conferring no conformance standing. `provisional` is not a registration status under §O6.

Universal Manifest Profile Registry PREVIEW

Companion to: Base specification. **Governed by:** EXT-OPT §O6. **Registry categories:** `signature`, `trust`, `domain`, `binding`.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC 2119] [RFC 8174] when, and only when, they appear in ALL CAPITALS, as shown here.

R1. Scope

This document prepares the registry that EXT-OPT §O6 requires. §O6 defines the registration *mechanism*; this document holds the draft *contents*. Explicit recognition in the Base or a normative

extension remains the authority for existing profiles. A draft listing does not activate an additional registration.

This registry is published independently of the specification version. A profile is registered, deprecated, or sunset by editing this document through the process in R4; no specification release is required and none is implied. Where the published `schema.json` enumerates values that correspond to entries here — today, the `algorithm` enumeration of the `signature` category — that enumeration is the **wire projection** of this registry, and this registry is the governing record.

An implementation MUST NOT infer that an identifier is registered because it appears in a specification draft, a design note, or a conformance fixture. Registration is constituted by an entry in this document with status `active`.

R2. Identifier Grammar and Categories

Every registered profile has a canonical identifier of the form:

```
um:profile:<category>:<name>
```

`<category>` MUST be exactly one of the four values below. `<name>` is a lowercase token using hyphens as word separators.

- **`signature`** — signature suites. A suite's identity is the (`algorithm`, `canonicalization`) pair; the identifier in this category names that pair.
- **`trust`** — trust, binding, and assurance profiles.
- **`domain`** — domain-specific manifest class profiles.
- **`binding`** — credential binding profiles.

Identifiers are globally unique. An identifier MUST NOT be reused after deprecation, and MUST NOT be reassigned to different semantics.

Profile identifiers are `@id`-coerced in the published JSON-LD context, so a registered identifier expands as a compact IRI under the version-independent `um:` prefix and is a globally unique IRI rather than a bare label. Receipt reason codes and consent condition values, which also route through this registry's process (R6), are **not** `@id`-coerced: they are inert literal strings compared as opaque tokens. Registrants MUST NOT assume the two behave alike.

R3. Entry Fields and Status Vocabulary

Each registry entry carries the six fields §O6 requires: **identifier**, **name**, **version**, **status**, **specification URI**, and **registration date**, with one of §O6's three statuses. The registry also lists **provisional identifiers**: names reserved against collision that are not registry entries. A provisional identifier is shown with the label `provisional` and "Not registered", and §O6's entry fields and statuses do not apply to it.

Status values:

- **`active`** — registered. The profile has completed R4, including at least one conformance fixture integrated into the suite. Only an `active` profile is registered for conformance purposes.
- **`provisional`** — named in the specification document set, but not registered. It has not completed R4 and MAY lack a conformance fixture. A `provisional` identifier confers **no** conformance standing, and an evaluator MUST NOT treat a `provisional` identifier as recognized. This status exists so that identifiers already visible in the specification set are reserved against collision and are publicly accounted for. *(Adopted by the maintainers for v0.5. `provisional` is a listing label for identifiers that are not registry entries; §O6's three statuses below apply to registry entries.)*
- **`deprecated`** — registered, superseded or withdrawn, and carrying a date and a superseding-profile reference where one exists. Evaluators SHOULD warn on a reference to a deprecated profile but MUST NOT reject a manifest solely for referencing one.
- **`sunset`** — withdrawn at a major-version boundary. Evaluators MAY reject.

For a `provisional` identifier, a specification URI given as a section reference rather than a stable published URL is not yet independently citable. A section reference in an `active` entry, such as Base §1.6 or EXT-OPT §O15, points to text that already recognizes the profile.

R4. Registration Procedure

In this document, **the maintainers** means the maintainers of the canonical `UniversalManifest/universal-manifest` repository, acting through its public process.

Registration follows the five steps of EXT-OPT §O6, restated here as the operative procedure:

1. **Proposal.** The author submits a specification to the maintainers as an issue or pull request on the canonical `UniversalManifest/universal-manifest` repository, including a scope

statement, normative requirements using RFC 2119 keywords, security considerations, an interoperability assessment, and **at least one conformance test fixture**.

2. **Review.** The maintainers review for completeness, security, and interoperability, and MUST verify that the proposed profile does not conflict with an existing profile at the same extension point.
3. **Conformance integration.** The author supplies the fixtures; the maintainers integrate them into the conformance suite.
4. **Registration.** On the maintainers' public acceptance, the profile receives its canonical identifier and its entry is added to this document with status `active`.
5. **Maintenance.** The author maintains the specification and its tests. The maintainers MAY deprecate a profile if maintenance lapses, a security issue arises, or a superseding profile is registered.

Conflict resolution. Two profiles defining incompatible semantics at the same extension point MUST NOT both be registered without resolution. The maintainers MUST resolve by merging them, selecting one, or defining an explicit compatibility boundary.

Registration policy. §O6 does not name a registration policy in the IANA sense, and this document does not invent one. Registration today requires the maintainers' public acceptance at step 4. Whether a lighter-weight path with designated expert review should exist for low-risk profiles is an open question recorded in §O6 and is **not** settled by this document.

R5. The Registry

R5.1 Category: `signature`

`um:profile:signature:jcs-ed25519`

- Name: Signature Profile A
- Version: 1.0
- Status: `active`
- Specification: Base §1.6 (Signature Profile A — JCS + Ed25519)
- Registered: 2026-08-08 (in use since v0.2)
- Notes: The mandatory-to-implement baseline. Every evaluator MUST support this profile. For wire `0.5`, the candidate applies all six predicates of Base §1.6.1.1: canonical non-identity

prime-order A/R, canonical S and the full-challenge uncofactored equation over exact signing bytes. Earlier specification versions retain their own acceptance rules. This is a version-bound constraint on the existing pair, not an additional suite registration; current-source/runtime and adversarial qualification remain required.

`um:profile:signature:jcs-es256`

- Name: Signature Profile B
- Version: 1.0
- Status: `active`
- Specification: EXT-OPT §O15 (Signature Profile B — JCS + ES256)
- Registered: 2026-08-08 (recognized in v0.4)
- Notes: ECDSA over NIST P-256 with SHA-256. Canonicalization and signing input are identical to Profile A. `signature.value` is the base64url of the raw 64-byte `R || S`; DER is rejected. The profile imposes **no** low-S rule and **no** RFC 6979 requirement, because hardware signers expose no nonce control. Optional-additive for evaluators. This profile exists so that hardware key custody — Secure Enclave, StrongBox, TPM — is reachable, since those chips hold P-256 keys only.

`um:profile:signature:jcs-ml-dsa-65`

- Name: JCS + ML-DSA-65 *(not registered in v0.5; see the maintainer decisions above)*
- Version: 0.1
- Status: `provisional`
- Specification: EXT-T2 §T2.3.2 (post-quantum signatures)
- Registered: not registered
- Notes: ML-DSA-65 with `JCS-RFC8785` canonicalization, carried in `postQuantumSignature` rather than `signature`. PREVIEW. The identifier is reserved here against collision; the profile has no conformance fixture and is not registrable until it does.

R5.2 Category: `trust`

`um:profile:trust:locked-tier-portable-unlock` — Version 0.1 — Status `provisional` — Specification: EXT-T1 §T1.3.3 — Not registered.

`um:profile:trust:zkp-bbs-linked-secret` — Version 0.1 — Status `provisional` — Specification: EXT-T2 §T2.1 — Not registered.

`um:profile:trust:tier2-privacy-preserving-binding` — Version 0.1 — Status `provisional` — Specification: EXT-T2 — Not registered.

`um:profile:trust:current-policy-state` — Version 0.1 — Status `provisional` — Specification: EXT-OPT §O2.6 — Not registered.

R5.3 Category: `domain`

`um:profile:domain:receipt` — Version 0.1 — Status `provisional` — Specification: EXT-OPT §O3 — Not registered.

`um:profile:domain:represented-person-consent` — Version 0.1 — Status `provisional` — Not registered.

`um:profile:domain:referral-route-attribution` — Version 0.1 — Status `provisional` — Specification: EXT-OPT §O13 — Not registered.

`um:profile:domain:geopose-primary-pose` — Version 0.1 — Status `provisional` — Not registered.

`um:profile:domain:personal-data-vault` — Version 0.1 — Status `provisional` — Not registered.

- Notes: **This identifier already appears on the wire but is defined in no specification section.** It is recorded here to make that gap visible and to reserve the identifier. It MUST NOT be treated as recognized until a defining section exists and it completes R4.

R5.4 Category: `binding`

`um:profile:binding:sd-jwt-kb` — Version 0.1 — Status `provisional` — Specification: EXT-T1 §T1.1 — Not registered.

`um:profile:binding:apple-devicecheck-token` — Version 0.1 — Status `provisional` — Not registered.

`um:profile:binding:webauthn-uv-platform` — Version 0.1 — Status `provisional` — Not registered.

R6. Subsidiary Value Registries

§O6 governs five subsidiary value registries in addition to the profile categories above. They share this document's registration process but are enumerated in their defining sections rather than restated here, because their contents change with the specification's wire format rather than independently of it.

- **Receipt event classes** — EXT-OPT §O3.2.
- **Receipt reason codes** — EXT-OPT §O3.3. Values are opaque literal strings, not IRIs.
- **Agent-delegation scope values** — EXT-OPT §O7.
- **Liveness `proofType` values** — EXT-T1 §T1.3 defines validation mechanics for the recommended `webauthn-uv` method. This draft is not evidence of a completed registration. A successfully validated proof may satisfy the specified liveness floor; the method name or a registry entry alone cannot validate a proof or establish hardware provenance.
- **Additional JWE algorithm pairs** — Base §2.4.

A value appearing in a design-stage draft is **not** registered. Appearance of a candidate token in a draft confers no context registration and no conformance standing.

R7. Maintenance

On publication under R4 at the address §O6 names, this document becomes the registry of record. Maintainer changes then take effect on publication, independently of any specification release. v0.5.0 carries this document as the local draft that §O6 describes; it has not performed that publication act.

Each change MUST preserve the append-only property of identifiers: entries may change status and version, but an identifier once published here is never reused for different semantics, and never removed.