

Inventory Hints Contract v1.0 Media Audit Tool™ / Asset Intelligence Engine™ 2026-07-06

Relationship to the Analyzer Fact Contract

This is a separate, sibling contract. The Analyzer Fact Contract v1.0 is frozen and unaffected. Facts describe references to specific assets (“this Download references this path”). Inventory hints describe scan scope — which directories an Analyzer owns and which subdirectories within its own territory are the Analyzer’s own interface furniture rather than user media. These are different concerns and do not belong in the same contract.

Why this exists

Some extensions (jDownloads is the motivating case) store their own bundled interface graphics — category icons, featured badges, file-type icon sets — inside /images/<extension>/ rather than /media/com_<extension>/. Because these files are pulled in by fixed filename convention rather than referenced from a database field, no Analyzer can produce Facts for them. Without a way to identify them, they surface as false-positive orphans, sometimes by the hundreds, and administrators can end up trashing an extension’s own working interface files.

The hook

onMediaaudittoolCollectInventoryHints()

Returns:

```
[
  'analyzer' => 'jdownloads',
  'included_roots' => [
    'images/jdownloads',
  ],
  'reserved_paths' => [
    'images/jdownloads/catimages',
    'images/jdownloads/featuredimages',
    'images/jdownloads/fileimages',
    'images/jdownloads/hotimages',
    'images/jdownloads/miniimages',
    'images/jdownloads/newimages',
    'images/jdownloads/updimages',
    'images/jdownloads/downloadimages',
  ],
]
```

included_roots — directories this Analyzer’s extension owns. Informational; may overlap with what the Analyzer already contributes via the existing onMediaaudittoolCollectIncludedDirectories hook.

reserved_paths — subdirectories within this Analyzer’s own territory that hold the extension’s own interface assets, not user media. These are the only paths this contract concerns itself with.

The security constraint

A hook that lets a plugin say “never flag files here” is a hiding spot if left unconstrained. A compromised or malicious Analyzer could declare something broad, or a directory it doesn’t actually own, as reserved — and MAT, a tool whose purpose is surfacing what’s unreferenced on a site, would go blind to it. Three rules close this off:

Containment by construction. A reserved_paths entry is only honored if it is nested under one of that same Analyzer’s own included_roots (or its existing Included Directories contribution). An Analyzer can carve exceptions out of its own claimed territory only — never reach outside it to reserve another extension’s directory, a core Joomla directory, or an arbitrary upload folder. MAT validates this and silently discards any hint that fails containment.

Minimum specificity. A reserved path must be a real, specific subdirectory. Shallow or global paths (images, images/jdownloads on its own) are rejected. This prevents a careless or malicious declaration from swallowing an entire root.

No hiding — only “don’t recommend deleting.” Reserved paths remain fully inventoried and visible. Nothing is removed from the audit. This is the core mitigation: an admin can always see everything, including every reserved file, its size, and which Analyzer claimed it. **The contract changes what MAT recommends, never what MAT shows.**

Inventory status buckets

Reserved is a first-class inventory status, sitting alongside the existing three:

Referenced — matched to a Fact from an Analyzer. Unreferenced — not matched to anything; a genuine orphan candidate. Reserved — matched to a declared reserved_paths entry; a known Analyzer-owned interface asset. Trashed — moved to Media Audit Tool’s trash.

Reserved behavior

Reserved assets are inventoried and visible in the Inspector, same as any other file. Reserved assets are not counted as Unreferenced and do not contribute to orphan totals. Reserved assets do not show bulk-selection checkboxes in trash workflows. If a reserved path is somehow submitted in a trash request (stale UI state, direct API call, etc.), server-side code rejects it. The UI omission is not the only safeguard. Preview/notes messaging reads: “Reserved interface asset.”

Transparency requirement

The full list of declared reserved paths, and which Analyzer declared each one, must be visible somewhere in the admin UI (Options page or Inspector). If a plugin ever declares something unusual, it is sitting in plain sight for an administrator to notice and question, not buried in code. Silent, invisible exclusion is exactly the failure mode this contract exists to prevent.

Rollout note

This contract should be implemented before general release for any Analyzer that bundles interface assets under inventoried media directories (for example, jDownloads), as otherwise those assets will be incorrectly presented as orphan candidates. This is no longer theoretical: testing has already surfaced hundreds of jDownloads interface icons flagged as false-positive orphans.

Future enhancement (not for v1.0)

The Inspector could eventually offer a status filter — Referenced, Unreferenced, Reserved, Trashed — so Reserved reads as a natural inventory state rather than a hidden implementation detail. Not needed now; noted here so it isn't lost.

A note on naming

This contract deliberately keeps the field name analyzer rather than a more generic term like provider, even though every current source of inventory hints happens to be an Analyzer plugin. This is an intentional choice to preserve the Analyzer nomenclature as the platform's marketing identity. If a future non-Analyzer source of inventory hints is ever needed, that can be addressed at that time without disrupting this contract.

Copyright 2026 In Color® <https://incolor.dev>