

# QA Standards for Media Asset Generation

Maintained by	ARIA — Symph Agency Intelligence Engine
Approved by	Xian Baylin (QA Owner)
Effective date	2026-02-24
Applies to	All marketing asset generation: images, videos, carousels
Generated	2026-02-25 13:48 UTC

---

## 1. Core Principles

Governing rules that apply to every asset, every time

### Preflight First

Resolve platform, placement, and dimensions BEFORE generating — never generate and validate later.

### No Silent Overrides

If a user requests a specific format, honour it exactly. If it is invalid, block with a clear reason.

### No Fake Completions

A lifecycle state cannot be set unless all required data is present (e.g. verdict\_ready requires a verdict and timestamp).

### Deliver Only on Pass

Users receive QA-passed assets only. Warning or Failed assets are fixed first, not delivered.

### Preserve Originals

Fix/Regenerate creates a new asset with a new ID. The original is never overwritten.

### Brand Per Product

Each product has its own approved brand. Colors, logos, and typography must never be mixed across products.

### Platform Layer is Independent

Brand QA and Platform QA run as two separate, independent layers. Failure in either blocks delivery.

### Queue = Short-Lived State

Assets in queued state must be picked up within 10 seconds. Anything queued >30s escalates automatically.

---

## 2. Pre-Generation Workflow (7 Steps)

Required steps before any asset is generated

### Step 1 — Lock Context

Confirm all required fields: Product, Platform, Placement, Asset Type, Channel Source, Request ID. If any field is missing → block and request correction.

### Step 2 — Load Standards

Load: (1) Design Phase Brand Guidelines (primary), (2) Product Marketing Branding Profile, (3) Uploaded branding references, (4) Latest verified Platform Standards. If a refresh fails → use last verified and mark standards\_refresh\_pending.

### Step 3 — Resolve Output Specs

Produce before generation: target aspect ratio, target pixel dimensions, allowed alternatives, safe-zone requirements, minimum quality requirements, platform-fit validation rules.

### Step 4 — Preflight Validation

All fields must pass: product linked, platform set, placement set, branding profile available, platform standards resolved, output dimensions set. Any failure → block generation with a clear reason.

### Step 5 — Pass Specs as Hard Constraints

Pass the resolved specs into the generator: exact ratio, target dimensions, safe-zone guidance, branding constraints. Do not use a generic prompt when platform and placement are known.

### Step 6 — Post-Generation Compliance Check

Immediately after generation: verify actual dimensions match spec. If mismatch → auto-resize if ratio is correct, else mark off\_spec and trigger Fix/Regenerate. Do not send a success signal on an off-spec asset.

### Step 7 — QA Verification

Run dual-layer QA (Brand Layer + Platform Layer) against the resolved spec. QA is a verification step — not a replacement for preflight.

---

## 3. Dual-Layer QA Enforcement

Two independent QA layers — both must pass before delivery

### Layer A — Brand QA

<b>Correct product name and branding</b> Product name, logo, and tagline match approved brand	<b>PASS</b>
<b>Approved color palette only</b> No cross-product color contamination	<b>PASS</b>
<b>Logo placement per brand spec</b> Logo at correct position, size, and contrast	<b>PASS</b>
<b>Typography matches brand spec</b> Correct typeface, weight, and sizing	<b>PASS</b>
<b>No invented brand elements</b> No unofficial variants, icons, or colors	<b>PASS</b>
<b>Tone and messaging match brand voice</b> Casual/formal/bold as per brand personality	<b>PASS</b>

### Layer B — Platform QA

<b>Dimensions match platform spec</b> Actual pixel dimensions = target dimensions	<b>PASS</b>
<b>Aspect ratio correct for placement</b> Feed, Story, Reel ratios enforced per platform	<b>PASS</b>
<b>Safe zones respected</b> No critical content in crop/UI-overlap zones	<b>PASS</b>
<b>Minimum resolution met</b> Output meets platform minimum quality requirements	<b>PASS</b>
<b>File format valid</b> PNG/JPG/MP4 appropriate for the placement	<b>PASS</b>
<b>No platform-prohibited content patterns</b> Text density, overlays, and contrast per platform rules	<b>PASS</b>

## QA Verdict Definitions

Verdict	Meaning	Action Required
■ Pass	Both layers clear. Asset is compliant.	Deliver to user
■ ■ Warn	Minor issue found. Asset is usable but suboptimal.	Fix before delivery (or flag with reason)
■ Fail	Critical issue found. Asset must not be delivered.	Trigger Fix/Regenerate loop
■ Error	QA could not complete (timeout, missing file, etc.).	Mark stalled, surface reason to operator

## 4. Platform Dimension Specifications

Hard generation constraints per platform and placement

Platform	Placement	Ratio	Target Dimensions	Notes
Facebook	Feed (Static)	1:1	1440 × 1440 px	Primary recommended
Facebook	Feed (Static)	4:5	1440 × 1800 px	More screen coverage
Facebook	Story / Reel	9:16	1080 × 1920 px	Full screen vertical
Instagram	Feed	4:5	1080 × 1350 px	Preferred — most coverage
Instagram	Feed	1:1	1080 × 1080 px	Square fallback
Instagram	Story / Reel	9:16	1080 × 1920 px	Full screen vertical
X (Twitter)	Feed	16:9	1200 × 675 px	Primary landscape
X (Twitter)	Feed	1:1	1080 × 1080 px	Square alternative
LinkedIn	Feed Post	1.91:1	1200 × 628 px	Standard link preview size

Platform	Placement	Ratio	Target Dimensions	Notes
LinkedIn	Feed Post	1:1	1080 x 1080 px	Square feed post
TikTok	In-Feed Video	9:16	1080 x 1920 px	Vertical only

■ Gemini image output is ~1024px — always auto-resize to target before QA when aspect ratio is correct. Auto-resized assets are compliant (off\_spec = false after resize).

## 5. Format Intent Enforcement

User-requested formats are high-priority constraints

Intent Status	Meaning	Behaviour
matched_user_intent	User specified a format; system honoured it.	Use exactly as requested
platform_default	No format specified; platform standard applied.	Apply platform spec automatically
adjusted_with_warning	User's requested format is invalid for platform.	Block generation; list allowed formats
fallback_applied	No exact spec match; closest fallback used.	Log warning; add QA flag

Every generated asset records intentMatchStatus, requestedRatio, and resolvedRatio in its metadata for full traceability.

## 6. Fix & Regenerate Rules

How to handle assets that fail or warn in QA

### Do NOT overwrite originals

Every fix creates a new asset with a new UUID and version number. The original record is preserved.

### Link old ↔ new

New asset carries: previousVersionId, fixSource, fixIssuesUsed, fixTriggeredAt, and version number.

### Use QA reasons as fix constraints

Wrong ratio → specify correct ratio. Brand mismatch → include correct palette. Logo missing → specify placement.

### Auto-run QA on fixed version

After regeneration, QA runs automatically. The fix loop continues until QA passes.

### Fix button visibility

Show Fix button on: Warning, Failed. Hide on: Pass, Approved, Pending QA, Generation Failed.

## 7. Lifecycle State Machine

Valid states and required data for each transition

State	Description	Required Fields
queued	Asset created; waiting for QA worker pickup.	assetId, filePath, queuedAt
queued_delayed	SLA warn: not picked up within 30 seconds.	queuedAt (age > 30s)
scanning	QA worker is actively processing this asset.	assetId, filePath
verdict_ready	QA complete with a real verdict.	qaStatus (not null/none) + verdictAt
stalled	Processing failed or SLA hard limit exceeded.	stallReason + stalledAt
legacy_unscored	Pre-pipeline asset; needs QA rescan.	lifecycleNote + migratedAt
delivered	Asset delivered to user.	verdict_ready criteria + deliveredAt

Integrity rule: a state transition is blocked if its required fields are absent. verdict\_ready without qaStatus or verdictAt → system marks stalled and logs the reason.

## 8. QA Worker SLA

Automatic processing guarantees for queued assets

Event	Target	Action on Breach
Asset enters queue	< 10 seconds	Worker picks up within next 5s poll cycle
Not picked up	> 30 seconds	State escalates to queued_delayed
Not completed	> 120 seconds	State escalates to stalled; reason logged
QA timeout	> 90 seconds	Mark stalled with reason: QA request timed out
Concurrent writes	Always	File lock + atomic rename prevents JSON corruption
Worker crash	On restart	PM2 auto-restarts; resumes from queued/queued_delayed assets

## 9. Non-Negotiable Guardrails

These rules cannot be bypassed under any circumstances

- Never use generic 16:9 for platform-specific image posts unless explicitly requested.
- Never bypass preflight when platform and placement are known.
- Never mark an off-spec asset as final without a warning or status flag.
- Never overwrite the original asset during a fix or regenerate operation.
- Never detach QA verdicts from their exact Asset ID and version.
- Never disable platform QA enforcement because a standards source is temporarily inaccessible — use last verified.

- Never include technical specs (dimensions, 'square', '1080x1080') in image generation prompts — format specs belong in pipeline logic, not prompt text.
- Never mix one product's branding elements (colors, logo, typography) into another product's assets.
- Never set verdict\_ready without both qaStatus and verdictAt populated.
- Never label legacy or pre-pipeline assets with live pipeline states (use legacy\_unscored or migration\_pending).

---

## 10. QA Notification Standards

Required communication after every asset generation

When	Message Required
Asset generated	"Your asset for [Product] has been generated and sent for QA review. QA verdict pending (~15s)."
QA Pass	"QA Verdict for [Product]: Pass ■ — Review at <a href="https://symphony.symph.co/marketing">symphony.symph.co/marketing</a> "
QA Warning	"QA Verdict for [Product]: Needs Review ■■ — Reason: [specific reason] — Review at ..."
QA Fail	"QA Verdict for [Product]: Failed ■ — Reason: [specific reason] — Review at ..."
Asset stalled	"QA Verdict for [Product]: Error ■ — QA could not complete. Reason: [reason]"

Guardrails: Never claim QA is done before it has run. Never fabricate reasons. Always include the Content Review link.