**MCPX‑KendoBridge — Rotate Telerik KendoReact License Runbook (Build‑Time Only)**

**Document:** runbooks/rotate\_telerik\_license.docx  
**Version:** 2.0.0 (Kendo Migration)  
**Last Updated:** 2025‑09‑27  
**Owner:** SRE Lead (Responsible) — DoSE (Accountable) — DocFactory (Author)  
**Applies to:** Admin Portal (KendoReact Fluent v12; ThemeBuilder overrides). CI builds only.

**Purpose.** Define a **safe, auditable** procedure to **rotate the Telerik KendoReact license** used during **UI build time** in CI. The license is **never** stored in code, images, or the database, and is **only** provided to CI via **GitHub Environments** secrets. This runbook ensures zero leakage, zero downtime, and clear evidence capture across **Alpha → Beta → RTM → Prod**.

**Compliance banner (always in effect):**

* **Add‑only** schema; **Stored‑procedure‑only** DAL; **No‑Hard‑Coding** of dynamic values (child cmd/args/cwd, timeouts, heartbeat cadence, allowed origins, feature flags).
* **Secrets** — including the Telerik license — **never** appear in code, DB, logs, or artifacts; they live **only** in **GitHub Environments** or vendor portals. The Admin Portal consumes the license at **build time only**, never at runtime.

**1) Scope & Roles**

* **Scope:** KendoReact Admin Portal builds (any repo path such as /web or /web/admin-portal).
* **Not in scope:** API service, database, or runtime configuration.
* **RACI**
  + **A**: DoSE
  + **R**: SRE Lead (build pipelines), Release Eng
  + **C**: SecLead, Dev Lead
  + **I**: QA, DocFactory

**2) When to Rotate**

Rotate the license when any of the following apply:

* Vendor lifecycle policy requires periodic rotation or renewal.
* Key **expiry** is approaching within 30 days.
* Personnel change or suspected exposure (treat as **P1** security event).
* New tenancy or environment isolation is mandated by compliance.

**3) Preconditions & Inputs**

* Valid new license material from the vendor portal.
* Access to **GitHub Environments** (alpha, beta, rtm, prod) with permission to set secrets.
* CI workflows reflect the **license-at-build** pattern (see §7 YAML).
* **Evidence Pack** folder structure and release draft exist (see docs/12\_evidence\_pack.docx).

**Secret names (standardized):**

* TELERIK\_LICENSE — multi‑line content (preferred), or
* TELERIK\_LICENSE\_PATH — path to the temporary license file created by CI.

**4) High‑Level Procedure (Zero Downtime)**

1. **Stage the new secret** in **Alpha** environment only.
2. **Build UI** in Alpha; verify logs **do not** contain the license; artifact compiles successfully.
3. **Promote secret** to **Beta → RTM → Prod** Environments after successful build/test at each stage.
4. **For Prod**, trigger a UI rebuild/redeploy through the standard CD job (no API downtime).
5. **Capture Evidence** (see §9) and update release notes.

**5) Detailed Steps (per Environment)**

**5.1 Update GitHub Environment Secret (Alpha)**

**Option A — GitHub UI**

1. Repo → **Settings** → **Environments** → alpha → **Secrets and variables**.
2. Add or update **Secret**: TELERIK\_LICENSE.
   * Paste the license content exactly as provided (multi‑line accepted).
   * Double‑check no trailing spaces; ensure **No logs/preview** are shown.
3. Save.

**Option B — CLI (preferred for audit trails)**

# NEVER echo the license to terminal; use a file or heredoc in a secure shell

gh secret set TELERIK\_LICENSE --env alpha --repo <org>/<repo> < license.txt

**Do not** commit license.txt anywhere. Delete the local file after setting: shred -u license.txt.

**5.2 Validate CI (Alpha)**

1. Dispatch or wait for a CI build that includes the **UI build** job.
2. Ensure the job step **“Prepare Telerik license file”** runs and sets TELERIK\_LICENSE\_PATH to a runner temp path.
3. Confirm **build success**; no warnings/errors about license; **no license content** in logs.
4. **Upload artifact** ui-dist; verify size/hash differences are normal.

**5.3 Promote to Beta → RTM → Prod**

Repeat §5.1 for each Environment, then trigger the build (and deploy for Beta/RTM/Prod) via the standard workflows.

* **RTM**: Proceed as usual; RTM remains **Prod DB read‑only** and is unaffected by UI licensing.
* **Prod**: Follow canary UI rollout (if applicable), then full rollout.

**6) Verification Checklist (per Environment)**

* CI **UI build** succeeded; no “license missing/invalid” messages.
* **Logs free of secrets** (license value masked; no raw content).
* ui-dist artifact present; no plaintext license within compiled assets.
* **Secret Scanning** shows no new findings.
* Evidence updated (see §9).

**7) CI Snippet (Authoritative License Injection Step)**

This step is already present in our **CI plan**; include it or keep it unchanged. It **never** logs the license and cleans up after the build.

# In .github/workflows/ci.yml, within the UI build job

- name: Prepare Telerik license file

if: ${{ steps.kendo.outputs.kendo == 'true' }}

env:

TELERIK\_LICENSE: ${{ secrets.TELERIK\_LICENSE }}

TELERIK\_LICENSE\_PATH: ${{ secrets.TELERIK\_LICENSE\_PATH }}

run: |

set -euo pipefail

# Mask any accidental echo

echo "::add-mask::$TELERIK\_LICENSE"

# Require at least one of the two inputs

if [ -z "${TELERIK\_LICENSE:-}${TELERIK\_LICENSE\_PATH:-}" ]; then

echo "Missing Telerik license secret for Kendo build." >&2; exit 1; fi

# If content provided, write it to a temp file

if [ -n "${TELERIK\_LICENSE:-}" ]; then

echo "$TELERIK\_LICENSE" > "$RUNNER\_TEMP/telerik-license.txt"

echo "TELERIK\_LICENSE\_PATH=$RUNNER\_TEMP/telerik-license.txt" >> "$GITHUB\_ENV"

fi

**Post‑build cleanup (optional hardening):**

- name: Cleanup Telerik license file

if: ${{ always() && steps.kendo.outputs.kendo == 'true' }}

run: |

test -f "$TELERIK\_LICENSE\_PATH" && shred -u "$TELERIK\_LICENSE\_PATH" || true

**8) Troubleshooting (Quick Matrix)**

| **Symptom** | **Likely Cause** | **What to Check / Fix** |
| --- | --- | --- |
| “License not found/invalid” during UI build | Secret absent or malformed | Secret exists in the **Environment** targeted by the workflow; no CR/LF issues; CI step writes TELERIK\_LICENSE\_PATH |
| Build passes locally but fails in CI | Env secret missing or different | Compare envs; ensure @progress/kendo-\* deps present and UI build job executed |
| License string appeared in logs | Accidental echo or verbose build | Ensure masking step in CI; set set +x; scrub logs; rotate license again |
| UI asset contains license text | Build mis‑configured | Search dist recursively; if found, stop, scrub artifacts, rotate immediately; open incident (P1) |
| Windows runners newline issues | CRLF handling | Use printf or a file upload; avoid PowerShell echo with newlines |

**9) Evidence Pack (What to Attach)**

Attach the following to the **Release** (retain **≥ 1 year**):

* CI job logs (scrubbed) showing successful **Prepare Telerik license file** and UI build.
* Proof of **no license content** in ui-dist (e.g., grep -R summary or artifact scan).
* Secret Scanning summary (no findings).
* Date/time and Environments where the secret was updated (screenshot of Environments page is acceptable).
* If rotation was triggered by expiry or exposure, include a **short note** in release notes and, if exposure, an **incident report**.

**10) Security & Compliance Controls**

* **Never** store or reference license material in: source files, DB, migration seeds, SP bodies, or runtime config.
* **Do not** ship license content inside container images or UI bundles.
* **Secret Scanning** must remain enabled on the repo/org; treat hits as **P1** until cleared.
* Rotation steps are **auditable** via GitHub Environment history and CI job logs (scrubbed).

**11) Rollback / Backout**

If UI build fails after updating the secret:

1. **Revert** to the previous secret value (GitHub Environment → secret history or paste former content).
2. Re‑run the **UI build** job; if still failing, open an incident and proceed with **Rollback Runbook** for the UI asset only (API unaffected).
3. Keep both attempts’ logs (scrubbed) in Evidence.

**12) Rotation Cadence & Ownership**

* **Cadence:** at least annually, or per vendor policy; immediately on suspected exposure.
* **Ownership:** SRE Lead maintains the schedule; SecLead validates; DoSE approves completion.
* **Audit:** Verify via Evidence Pack and a checklist issue in the release milestone.

**13) Appendices**

**A) Secret Update (CLI) — Examples (non‑secret)**

# Provide the new license as a local file; NEVER commit it

gh secret set TELERIK\_LICENSE --env alpha --repo <org>/<repo> < license.txt

gh secret set TELERIK\_LICENSE --env beta --repo <org>/<repo> < license.txt

gh secret set TELERIK\_LICENSE --env rtm --repo <org>/<repo> < license.txt

gh secret set TELERIK\_LICENSE --env prod --repo <org>/<repo> < license.txt

shred -u license.txt

**B) Dist Scan (ensure no plaintext license)**

# After CI uploads ui-dist, scan quickly (local or in a secured runner)

grep -R --binary-files=without-match -n "Telerik" dist/ || true

**C) References**

* **CI/CD Plan:** see docs/10\_ci\_cd.docx (UI build & license guard).
* **Compliance:** see docs/13\_compliance.docx (secrets policy, CSP/egress, DB rules).
* **Evidence Pack:** see docs/12\_evidence\_pack.docx.
* **Rollback:** runbooks/rollback.docx (UI asset flip if needed).
* **Monitoring:** docs/11\_monitoring.docx (not directly impacted by build‑time license).

**14) Assumptions**

1. The Admin Portal depends on @progress/kendo-\* packages and requires a valid license at **build time** only.
2. GitHub Environments (alpha, beta, rtm, prod) exist and are the **only** place secrets live for this project.
3. **RTM** uses **Prod DB (read‑only)** for parity; UI licensing does not access DB or runtime secrets.

**15) Next Steps**

* Add a **yearly rotation reminder** (GitHub scheduled issue) and verify **no leakage** via periodic artifact scans.
* Keep the **CI license step** identical across repos; do not fork bespoke logic.
* Include this runbook link in the PR template’s **No‑Hard‑Coding / Secrets** checklist.

**Footer (optional for Word header/footer):**  
*MCPX‑KendoBridge • Rotate Telerik License • v2.0.0 • 2025‑09‑27 • Confidential — Technijian Internal*