File Explorer Legacy Ribbon Fallback — Incident Report & After-Action Review

Report date: September 10, 2026 **Machine:** Windows 11 25H2, x64 (build 26200.9278 → reverted to 26200.9168) **Status:** File Explorer resolved; component-store corruption (OOBE payloads) remains — monitoring **Prepared from:** live elevated diagnostic session, Sep 9 23:49 – Sep 10 00:35 local time

---

1. Executive Summary

On September 8–9, 2026, File Explorer on this Windows 11 25H2 machine began rendering the **legacy Windows 10 ribbon interface** — no tabs, no modern command bar, no address/search band. The behavior persisted across Explorer restarts, a reboot, and Safe Mode.

The September cumulative update **KB5120998 (build 26200.9278)** had partially failed during its **online AppX servicing phase** with `0x80070570` (ERROR_FILE_CORRUPT) at Sep 8 22:00, in the same window as ~51 NTFS file-system index corruption events. This left the component store damaged ("repairable") and the XAML File Explorer host unable to initialize, so Windows fell back to the legacy shell by design.

Repair attempts could not fix the store: SFC reported no integrity violations (live files matched the store), and `DISM /RestoreHealth` failed with `0x800f0915` ("repair content could not be found anywhere") even after the download was un-stuck. The issue was resolved by **uninstalling KB5120998** (reverting to build 26200.9168) and rebooting. The modern File Explorer was verified back afterward.

A post-resolution store audit found the corruption is **localized**: 766 corrupt payload files, all inside the **OOBE component** (`userexperience-oobe 10.0.26100.1591`). The same `RestoreHealth` failure is recorded as far back as **July 23, 2026**, so the store had been unrepairable via Windows Update for weeks. This pre-existing corruption is the likely reason the LCU's OOBE package update failed and triggered the fallback; it has low day-to-day impact but can block future cumulative updates.

---

2. Symptom Detail

- File Explorer windows showed the Windows 10-style **ribbon** (`UIRibbonCommandBar`, `UIRibbonWorkPane`), plus `ReBarWindow32`, `TravelBand`, `UpBand`.
- No **tabs**, no **address/search band**, no **modern command bar**.
- Persisted after killing/restarting `explorer.exe`, after a full reboot, and in **Safe Mode** — a system-level shell issue, not a per-user setting.
- Separately, `explorer.exe` was crash-looping in `SHELL32.dll` (`0xc0000005`, WER bucket `bd4a2466…af452332`, offset `0x38d4f`) on Sep 9 16:05–16:06. This was mitigated earlier by unregistering the File Shredder shell extension (`fsshell.dll`, CLSID `{736AF091-C361-49B4-A928-87C586130D33}`) and deleting `HKCU\...\ShellBrowser\ITBar7Layout`; the crash loop stopped, but the ribbon UI remained.

Window-class evidence (legacy shell, Sep 9)

`CabinetWClass` contained the legacy shell composition:

`Address Band Root`, `Breadcrumb Parent`, `DirectUIHWND`, `DUIViewWndClassName`, `msctls_statusbar32`, `NamespaceTreeControl`, `NetUIHWND`, `NUIPane`, `ReBarWindow32`, `Search Box`, `SearchEditBoxWrapperClass`, `SeparatorBand`, `SHELLDLL_DefView`, `ShellTabWindowClass`, `SysTreeView32`, `ToolbarWindow32` (Navigation buttons / Up band / Address band), `TravelBand`, **`UIRibbonCommandBar`**, `UIRibbonCommandBarDock` ×4, **`UIRibbonWorkPane` ("Ribbon")**, `UniversalSearchBand`, `UpBand`, `WorkerW`.

---

3. Environment / Baseline

Item	Value
OS	Windows 11 25H2, x64, build 26200 (UBR 9278 at incident; 9168 after rollback)
Registry `ProductName`	"Windows 10 Pro" (stale branding — cosmetic only)
Disks	NVMe `KXG60ZNV512G TOSHIBA` (512 GB, disk 0); NVMe `SKHynix_HFS001TDE9X084N` (1 TB, disk 1) — both Healthy, BusType **RAID**
Storage controller	Intel RST VMD Controller 9A0B / VMD Managed Controller 09AB — driver **18.7.6.1010 (Intel, 2022-09-19)**
LCU chain (CBS)	26100.1742.1.10 → 26100.8737.1.24 → 26100.8875.1.28 → 26100.9168.1.19 (Aug 22) → **26100.9278.1.35 (Sep 8, KB5120998)**
Shell binaries (live)	explorer.exe 8875, Windows.FileExplorer.Common.dll 8875, Windows.UI.Immersive.dll 8875, Windows.UI.Xaml.dll 8972, shell32.dll 9278, twinui.pcshell.dll 9278
`Windows.UI.FileExplorer.dll`	1,808,384 bytes, LastWrite Sep 8 21:28:24, SHA-256 `5063F5BBDAF05C00064FA12B1A955DD4DFA35D5C55DFABFD9AADBCBB1C2EBC7F`
Key shell packages	`MicrosoftWindows.Client.FileExp` 1000.26100.10.0; `Client.CBS` 1000.26100.360.0; `Client.Core` 1000.26100.133.0; `Client.CoreAI` 1000.26100.9278.0; `Client.OOBE` 1000.26100.55.0; `Microsoft.UI.Xaml.CBS` 9.2603.18001.0; `WindowsAppRuntime.CBS.2` 2.2.0.100

---

4. Detailed Timeline (local time)

Phase 0 — Baseline (before Sep 8)

Time	Event
Jul 23, 04:15	First recorded `DISM /RestoreHealth` failure `0x800f0915` (store already unrepairable via Windows Update)
Aug 8	LCU 26100.8875.1.28 installed
Aug 22, 20:05	LCU 26100.9168.1.19 installed (stable baseline; Explorer working)

Phase 1 — Onset (Sep 8)

Time	Event
14:28	**KB5120998 (build 26200.9278) installed** (DISM install time)
21:00–22:01	**NTFS Event 55 ×51** — "corruption was discovered in the file system structure on volume C: … file reference number `0x9000000000009`" (index structure). Security-descriptor/integrity streams (`:$SDH`, `:SII`) involved. Newest event 22:01:43; none afterward
21:28:24	Servicing writes: shell binaries/packages updated on disk (e.g., `Windows.UI.FileExplorer.dll`, FileExp package files)
22:00:19–22:00:22	FileExp pre-registration checks OK (`statusFound: true` ; logon registration result `0x0`)

Time	Event
22:00:28–22:00:38	TWinUI Event 5961: `MicrosoftWindows.Client.CBS` activations of **InputApp** and **CortanaUI** fail — "The file or directory is corrupted and unreadable"
22:00:46–22:00:47	AppX Event 468/605/401/404: **`0x80070570` / `0x80073CF9`** — "Failed to set access rights to…", "Failure in the Staged state handler" for `MicrosoftWindows.Client.OOBE_1000.26100.55.0` and `MicrosoftWindows.61869722.Speion_1000.26100.9278.0`
22:00:47	`Microsoft.Windows.StartMenuExperienceHost_10.0.26100.4768` StateLocations operation fails `0x80070570`
22:00:49–22:01:04	`MicrosoftWindows.Client.CBS` update **344.0 → 360.0** repeatedly fails at PrerequisitesChecked (`0x80070570`), for both the user SID and `S-1-5-18`
22:00:47–22:01:12	Repeated Event 860: "Failed to notify RDS of recovery trigger with error `0x80070570`"
22:01:03–22:01:11	`Microsoft.AAD.BrokerPlugin` repair/remove operations fail `0x80070570`
22:01:43	AppX service lifecycle transitions to **Stopping/Stopped**
22:00:29	Windows Update reports failure **`0x80070570`**
(later) Sep 9 22:54	Ongoing package-resource corruption: repair of `MdOdrMcpFilterPackage` fails `0x80073B1F` — "**ResourceMap Not Found … Verify that the package's 'resources.pri' file in the package is valid**"

Phase 2 — Persistence (Sep 9)

Time	Event
03:30	`volmgr` Event 46: crash dump initialization failed
03:39	**Reboot** — legacy ribbon Explorer becomes persistent
16:05–16:06	`explorer.exe` crash loop in `SHELL32.dll` (`0xc0000005`) — 4 crashes
~16:30	Mitigation: File Shredder `fsshell.dll` CLSID unregistered (backup `fs-clsid-backup.reg`); `ITBar7Layout` deleted (backup `shellbrowser-backup.reg`); crashes stop, ribbon remains

Phase 3 — Diagnostics (Sep 9, evening)

Time	Event
earlier	`DISM /CheckHealth` → "No component store corruption detected"; COM activation tests; file/hash/registry checks
23:49	Window-class dump confirms legacy `UIRibbonCommandBar` shell; XAML modules are loaded into Explorer but not driving the UI
23:53:01	Elevated scan launched: `DISM /ScanHealth` + `SFC /scannow`
23:56:30	**`DISM /ScanHealth`** → **"The component store is repairable"** (store corruption confirmed)
23:58:19	`SFC /scannow` → "Windows Resource Protection did not find any integrity violations" (live files match the store)

Phase 4 — Repair attempts (Sep 9 23:59 – Sep 10 00:22)

Time	Event
23:59:00	Repair chain launched: `chkdsk /scan` → `DISM /RestoreHealth` → re-verify
00:01:11	`chkdsk C: /scan` clean: 498,984,959 KB total, 1,299,841 files, 383,845 indexes, **0 KB in bad sectors**, 2.09 min
00:03–00:13	`RestoreHealth` downloads payloads (2,338.7 MB cached: KB5120998 `.esd`, KB5043080 `.wim`); **stalls at 48/100**; WU restarts download (events 44 at 00:05:02 and 00:11:14); no active Delivery Optimization job

Time	Event
00:14:54–00:15:13	Windows Update stack nudged (restart wuauserv / BITS / DoSvc)
00:15:55	CBS resumes: "WULib Mode Complete: [0]"
00:17:16	`RestoreHealth` **fails:** `0x800f0915` **— "The repair content could not be found anywhere."**
00:19:51	`ScanHealth` re-check → still "The component store is repairable"
00:21:36	`SFC /scannow` re-check → still no integrity violations

Phase 5 — Resolution (Sep 10)

Time	Event
00:22:11	`DISM /Remove-Package` launched for `Package_for_RollupFix~31bf3856ad364e35~amd64~~26100.9278.1.35`; CBS mapped state `Installed`; removal initiated
00:26:28	Removal completes → exit **3010** (success, reboot required); `RebootPending = True`; CBS: 9278 → pending removal, 9168 → pending activation
~00:30	**Reboot**
after boot	Build **26200.9168**; `RebootPending = False`; modern File Explorer confirmed

Phase 6 — Verification (Sep 10, post-reboot)

Check	Result
Build	`UBR = 9168`, DisplayVersion 25H2
Pending reboot	False
Active package	`…26100.9168.1.19`
Explorer window classes	`Microsoft.UI.Content.DesktopChildSiteBridge`, `TITLE_BAR_SCAFFOLDING_WINDOW_CLASS`, `InputNonClientPointerSource`, `InputSiteWindowClass` — **no ribbon classes**

Phase 7 — Post-resolution store audit (Sep 10)

Time	Event
00:42–00:43	System restore point **#273** created: "Stable state after KB5120998 rollback (File Explorer fixed)"
00:52:02–00:54:41	`DISM /ScanHealth` re-run → **"The component store is repairable"** — corruption survived the rollback
00:54:40	CBS detail: **766 × CSI Payload Corrupt**, all in `amd64_userexperience-oobe_31bf3856ad364e35_10.0.26100.1591_none_54983d2d7a4908c4` (`r\OOBE\resources.pri`, `WebExperienceHost.dll`, `WebHost.winmd`, `webapps\guidedsetup\...\area-content.local.json`); manifest/metadata corruption = 0; repaired = 0

5. Root Cause Analysis

5.1 Failure chain

1. **KB5120998's online servicing phase failed partway** at Sep 8 22:00 with `0x80070570` (ERROR_FILE_CORRUPT) across multiple system AppX packages, in the same window as NTFS index-corruption events.
2. The failure left the **AppX/servicing registration state inconsistent** and the **component store damaged** ("repairable").
3. The **XAML File Explorer host** (`MicrosoftWindows.Client.FileExp` → `FileExplorerManaged` extension) could not initialize from that state.
4. Windows fell back to the **legacy explorer shell** — a designed fallback, which is why the UI looked like Windows 10 even though `explorer.exe` and `Windows.UI.FileExplorer.dll` were present, correctly registered, signed, and loading.

5.2 Why it looked like "corruption of explorer.exe" (but wasn't)

- `explorer.exe` (8875) and the XAML DLLs loaded successfully; the window just used the legacy composition.
- `SFC` saw no integrity violations because the **live files matched the (damaged) store**.
- `DISM /ScanHealth` was the only tool that surfaced the real problem: **the component store itself was repairable/damaged**.

5.3 Suspected trigger

Transient storage/servicing corruption during heavy update I/O, likely aggravated by the **outdated Intel RST VMD driver 18.7.6.1010 (2022-09-19)**. The NTFS Event 55 burst occurred precisely during the update write window, and the subsequent AppX failures were all I/O-level (`FILE_CORRUPT`, "failed to set access rights").

5.4 Why the store stayed broken

`RestoreHealth` relies on Windows Update to fetch clean payloads. On this build, that path failed with `0x800f0915` ("repair content could not be found anywhere") after a stalled download — a failure mode also reported by other users on 26200.9278/26300.9278. A **local source (ISO)** or **in-place repair upgrade** is required to repair the store.

5.5 The store corruption is localized (post-resolution audit)

A fresh `ScanHealth` after the rollback (Sep 10 00:54) still reported "repairable". CBS itemized it:

- **766 × CSI Payload Corrupt**, all in one component: `amd64_userexperience-oobe_31bf3856ad364e35_10.0.26100.1591_none_54983d2d7a4908c4` — the **OOBE (out-of-box experience)** payload:
  - `r\OOBE\resources.pri`
  - `r\OOBE\WebExperienceHost.dll`
  - `r\OOBE\WebHost.winmd`
  - many `r\OOBE\webapps\guidedsetup\network\area-content\<locale>\area-content.local.json`
- Manifest corruption: **0**; metadata corruption: **0**; repaired: **0** (ScanHealth detects only).
- **The corruption predates this incident:** `C:\Windows\Logs\DISM\dism.log` records a `RestoreHealth` failure with `0x800f0915` on **July 23, 2026 04:15** — the store was already unrepairable via Windows Update weeks earlier.

Significance: the AppX package that failed first and hardest during the LCU (`MicrosoftWindows.Client.OOBE`) is built from this corrupt component — strong evidence that the pre-existing OOBE corruption caused the LCU's online phase to fail and, in turn, the Explorer fallback. Day-to-day impact is low (OOBE runs only during setup/reset), but it can block cumulative updates that touch OOBE and break Reset/Recovery scenarios.

---

6. Evidence Log

6.1 NTFS (Sep 8 21:00–22:01)

- Event ID **55**, ×51: "A corruption was discovered in the file system structure on volume C:. A corruption was found in a file system index structure. The file reference number is `0x9000000000009`."
- `chkdsk /scan` later clean (0 bad sectors) → not physical media failure; consistent with online self-healing after the servicing storm.

6.2 AppX deployment (Sep 8 22:00–22:01)

- Event **468**: `error 0x80070570: Failed to set access rights to <package>` (OOBE, Speion).
- Event **605**: `Error 0x80070570: Failure in the Staged state handler` / `PrerequisitesChecked state handler`.
- Event **401/404**: Register failed; specific text "**The file or directory is corrupted and unreadable**" (`0x80073CF9` wrapper).
- Event **685**: `StateLocationsOperationImplementation failed` for `StartMenuExperienceHost` (`statecreationhelpers.cpp,116,80070`).
- Event **860**: `Failed to notify RDS of recovery trigger with error 0x80070570` (repeated).
- **22:01:43**: Appxsvc lifecycle → Stopped.
- **Sep 9 22:54:49**: `MdOdrMcpFilterPackage` repair fails `0x80073B1F` — `ResourceMap Not Found … Verify that the package's 'resources.pri' file in the package is valid`.

6.3 TWinUI activation

- Event **5961** (Sep 8 22:00:38): `MicrosoftWindows.Client.CBS_cw5n1h2txyewy!InputApp` and `!CortanaUI` — "The file or directory is corrupted and unreadable. Activation phase: Deployment."
- LockApp "Access is denied" entries also present (common/noise).

6.4 Integrity tooling results

Tool	Result
`DISM /CheckHealth` (earlier)	No component store corruption detected
`DISM /ScanHealth` (23:56)	**The component store is repairable**
`SFC /scannow` (23:58, 00:21)	No integrity violations (both runs)
`chkdsk /scan` (00:01)	Clean; 0 KB bad sectors
`DISM /RestoreHealth` (00:17)	**Failed** `0x800f0915` — repair content not found
`DISM /ScanHealth` (00:54, post-rollback)	**Still "repairable"** — 766 CSI payload corruptions in `userexperience-oobe`

6.5 File / package integrity

- `Windows.UI.FileExplorer.dll`: hash matches WinSxS hardlink copy `…_10.0.26100.9278_none_22d2198b4b2baf0c\Windows.UI.FileExplorer.dll`; Authenticode **Valid**, signer **CN=Microsoft Windows**; loads via `NativeLibrary.Load`.
- Version-resource quirk: live file reports stub `10.0.10011.16384` while the store path reports `10.0.26100.9278` — normal for these stubs (same on `Windows.UI.Xaml.Controls.dll`), **not** corruption.
- MUI: `System32\en-US\Windows.UI.FileExplorer.dll.mui` 5,120 B (Sep 8 21:28, stub ver); shell32/twinui `.mui` = 9278; `Windows.UI.Xaml.mui` = 8972.
- FileExp package contents (all LastWrite Sep 8 21:28): `FileExplorerExtensions.dll` 6,372,864 B; `FileExplorerManaged.dll` 23,040 B; `OSClient.NET.dll` 50,567,168 B; `resources.pri` 1,335,064 B;

`appxmanifest.xml` 19,982 B. Manifest declares extension category **`com.microsoft.windows.extensions.xaml.fileexplorermanaged`** (ExtensionId `FileExplorerManaged`) and in-process COM servers.

- `HKCR\Extensions\…\PackageId\MicrosoftWindows.Client.FileExp…` — 9 registrations present.
- At the time of the failure, Explorer (PID 10088) had `Windows.UI.FileExplorer.dll`, `Windows.UI.Xaml.dll`, and `Microsoft.UI.Xaml.dll` (from `Microsoft.UI.Xaml.CBS_8wekyb3d8bbwe` and `Microsoft.WindowsAppRuntime.CBS`) loaded — yet rendered the legacy UI (runtime init failure → fallback).

6.6 COM & registry checks

- HKLM `InProcServer32` for `{2aa9162e-c906-4dd9-ad0b-3d24a8eef5a0}` and `{6480100b-5a83-4d1e-9f69-8ae5a88e9a33}` → `C:\Windows\System32\Windows.UI.FileExplorer.dll`, `ThreadingModel=Apartment`; no `LocalServer32`.
- Activation: `{6480100b}` → OK (`System.__ComObject`); `{2aa9162e}` → `E_NOTIMPL (0x80004001)`.
- No HKCU CLSID overrides; `{d93ed569-3e0f-4b5a-8358-116f2eeb2fe7}` absent; no `Windows.UI.FileExplorer.dll_` remnants.
- `Shell Extensions\Approved`: no entries for either CLSID. `Blocked`: only benign `{9421DD08-935F-4701-A9CA-22DF90AC4EA6}` (Epson Easy Photo Print 2). Toggle CLSID `{e2bf9676-5f8f-435c-97eb-11607a5bedf7}` not present.
- Policies\Microsoft\Windows\Explorer (HKLM/HKCU) contain no relevant values; HKCU Advanced only `TaskbarAutoHideInTabletMode=0`, `SeparateProcess=0`.
- FeatureManagement overrides: no Explorer/XAML enable/disable found — the fallback is **not** velocity-flag-gated.

6.7 Window classes — before vs. after

Legacy (Sep 9)	Modern (Sep 10)
`UIRibbonCommandBar`, `UIRibbonWorkPane` ("Ribbon"), `UIRibbonCommandBarDock` ×4	`Microsoft.UI.Content.DesktopChildSiteBridge`
`ReBarWindow32`, `TravelBand`, `UpBand`, `SeparatorBand`	`TITLE_BAR_SCAFFOLDING_WINDOW_CLASS`
`ToolbarWindow32` ×3, `Address Band Root`, `Breadcrumb Parent`	`InputNonClientPointerSource`, `InputSiteWindowClass`
`Search Box`, `SearchEditBoxWrapperClass`, `UniversalSearchBand`	(modern equivalents inside XAML island)

---

7. What Was Ruled Out

Hypothesis	Finding	Verdict
Classic-ribbon registry hack	No HKCU CLSID overrides; HKLM COM registrations correct; no `…dll_` paths	Ruled out
Corrupt `Windows.UI.FileExplorer.dll`	SHA-256 matches store; signature valid; loads	Ruled out
Live OS file damage	SFC clean (twice)	Ruled out
Physical disk failure	Both NVMe SSDs Healthy; chkdsk 0 bad sectors	Ruled out
User profile / user-specific state	Persisted in Safe Mode; not profile-bound	Ruled out
Third-party shell extension (ribbon)	Blocked list only Epson; File Shredder caused *crashes*, not the ribbon	Ruled out for ribbon
Feature flags / velocity	No Explorer/XAML override	Ruled out

Hypothesis	Finding	Verdict
Windows Update corruption of live files	Live files verified vs store; issue is store/servicing state	N/A

8. Resolution — Commands & Results

All commands run elevated:

```
:: 1. Verify filesystem
chkdsk C: /scan

:: 2. Verify component store (found damage)
dism /online /cleanup-image /scanhealth

:: 3. Attempt repair (failed — payloads unavailable)
dism /online /cleanup-image /restorehealth

:: 4. Remove the September LCU
dism /online /remove-package
/packagename:Package_for_RollupFix~31bf3856ad364e35~amd64~~26100.9278.1.35 /quiet /norestart
/LogPath:"%TEMP%\opencode\dism-remove-lcu.log"

:: 5. Reboot, then verify
```

Step	Result
`chkdsk /scan`	Clean
`ScanHealth`	Repairable (damaged)
`RestoreHealth`	`0x800f0915` (content unavailable)
`Remove-Package`	**Exit 3010** (success, reboot required)
Reboot	Build 26200.9168; modern shell restored

9. Verification (post-reboot)

- `UBR = 9168`, `RebootPending = False`, package `…26100.9168.1.19` active.
- Explorer PID 13172; window class dump shows the modern XAML shell and **zero** legacy ribbon classes.
- User confirmed visually: tabs/command bar/address bar are back.

10. Open Items & Recommendations

1. **Update the Intel RST VMD driver (recommended, deferred by user).** Current `18.7.6.1010` (2022-09-19). This is the leading candidate for the servicing-time corruption. Create a restore point first; the update may require a reboot and carries some risk on RAID/VMD systems.
2. **Component store corruption remains — localized to OOBE payloads.** 766 corrupt files in `userexperience-oobe 10.0.26100.1591` (`OOBE\resources.pri`, `WebExperienceHost.dll`, `WebHost.winmd`, guided-setup locale JSONs). Present since at least **July 23, 2026**. Windows Update cannot repair it (`0x800f0915`). Fixes, in order of invasiveness:
    - **Sourced repair:** mount a Windows 11 25H2 ISO and run `DISM /Online /Cleanup-Image /RestoreHealth /Source:WIM:<drive>:\sources\install.wim:<index> /LimitAccess`.

    - **In-place repair upgrade:** definitive; rebuilds the store and keeps files/apps. Risk of leaving it: future cumulative updates that touch OOBE may fail the same way this one did.
3. **September security fixes are not installed** (KB5120998 removed). Options: stay on 9168 until the next cumulative update, or reinstall KB5120998 after the VMD driver update and monitor.
4. **Recurrence plan:** in-place repair upgrade from a matching-or-newer ISO (keeps files and apps) — the alternate documented fix for this regression.
5. **Artifacts retained** (see Appendix A) in case of recurrence.

---

11. Lessons Learned

- A failed LCU **online AppX phase** can leave the shell in fallback while all binaries appear healthy — **SFC will not catch it**; `DISM /ScanHealth` is the diagnostic that does.
- Safe Mode persistence + profile independence = system-level shell component issue.
- WU-sourced `RestoreHealth` can fail on specific builds (`0x800f0915`); have a local-source/ISO fallback ready.
- A stalled WU download can often be revived by restarting `wuauserv`/BITS/DoSvc (it resumed CBS here), but if the content is genuinely unavailable, it will still fail.
- Outdated storage drivers are a plausible root trigger for transient corruption during heavy servicing I/O — keep RST/VMD current.
- Component-store corruption can **predate** an incident by weeks and stay invisible until an update touches the damaged component — check `CBS.log` / `dism.log` history, not just the current state.

Appendix A — Artifacts & Log Locations

Item	Location
DISM ScanHealth (initial)	`%TEMP%\opencode\dism-scanhealth.log`
SFC (initial)	`%TEMP%\opencode\sfc-scannow.log`
chkdsk	`%TEMP%\opencode\chkdsk-scan.log`
RestoreHealth	`%TEMP%\opencode\dism-restorehealth.log`
ScanHealth (2nd)	`%TEMP%\opencode\dism-scanhealth2.log`
SFC (2nd)	`%TEMP%\opencode\sfc-scannow2.log`
LCU removal	`%TEMP%\opencode\dism-remove-lcu.log`
ScanHealth (post-rollback)	`%TEMP%\opencode\dism-scanhealth3.log`
Store re-check script	`%TEMP%\opencode\storecheck.cmd`
Restore point script/log	`%TEMP%\opencode\restorepoint.ps1`, `%TEMP%\opencode\restorepoint-log.txt`
Status files	`%TEMP%\opencode\repair-status.txt`, `uninstall-status.txt`, `nudge-status.txt`
Scripts	`%TEMP%\opencode\scans.cmd`, `repair.cmd`, `nudge.cmd`, `uninstall-lcu.cmd`, `post-reboot-check.ps1`, `uiinspect.ps1`
CBS log	`C:\Windows\Logs\CBS\CBS.log`
DISM log	`C:\Windows\Logs\DISM\dism.log`

Item	Location
AppX deployment events	Event log `Microsoft-Windows-AppXDeploymentServer/Operational`
TWinUI events	Event log `Microsoft-Windows-TWinUI/Operational`
Backups	`%TEMP%\opencode\fs-clsid-backup.reg`, `shellbrowser-backup.reg`

Appendix B — Raw Event Excerpts

```
NTFS Event 55 (Sep 8 21:00–22:01, ×51):
  "A corruption was discovered in the file system structure on volume C:.
   A corruption was found in a file system index structure.
   The file reference number is 0x9000000000009."

AppX Event 404 (Sep 8 22:00:46):
  "AppX Deployment operation failed for package
   MicrosoftWindows.Client.OOBE_1000.26100.55.0_x64__cw5n1h2txyewy with error
   0x80073CF9. The specific error text for this failure is: The file or directory
   is corrupted and unreadable."

AppX Event 401 (Sep 8 22:00:51):
  "Deployment Register operation ... on Package
MicrosoftWindows.Client.CBS_1000.26100.360.0_x64__cw5n1h2txyewy
   failed with error 0x80070570."

TWinUI Event 5961 (Sep 8 22:00:38):
  "Activation for MicrosoftWindows.Client.CBS_cw5n1h2txyewy!InputApp failed.
   Error code: The file or directory is corrupted and unreadable.
   Activation phase: Deployment"

DISM RestoreHealth (Sep 10 00:17:16):
  "Error: 0x800f0915
   The repair content could not be found anywhere."

DISM Remove-Package (Sep 10 00:26:28):
  "DISM_REMOVE_EXIT 3010"  (ERROR_SUCCESS_REBOOT_REQUIRED)

CBS ScanHealth corruption detail (Sep 10 00:54:40):
  (p) CSI Payload Corrupt (n) amd64_userexperience-
oobe_31bf3856ad364e35_10.0.26100.1591_none_54983d2d7a4908c4\r\OOBE\resources.pri
  (p) CSI Payload Corrupt (n) ...\r\OOBE\WebExperienceHost.dll
  (p) CSI Payload Corrupt (n) ...\r\OOBE\WebHost.winmd
  (p) CSI Payload Corrupt (n) ...\r\OOBE\webapps\guidedsetup\network\area-content\<locale>\area-
content.local.json
  Total Detected Corruption: 766   (CSI Payload Corruption: 766; manifest/metadata: 0)
  Total Repaired Corruption: 0

DISM log (Jul 23, 2026 04:15:41) — prior unrepairable state:
  "Failed to restore the image health. (hr:0x800f0915)"
```

Appendix C — Corroborating Reports

- Microsoft Q&A — identical symptom on 25H2 26200.8655 after an LCU; fixed by uninstalling the update: https://learn.microsoft.com/en-us/answers/questions/5652474/when-i-open-file-explorer-i-no-longer-can-use-a-ta

- Broadband Bulletin — "File Explorer doesn't have tabs anymore" (25H2); Microsoft support could not fix via registry/PowerShell; resolved by in-place repair upgrade: https://broadbandbulletin.com/d/9647-solved-file-explorer-doesnt-have-tabs-anymore
- Reddit — KB5120998 issues including DISM `RestoreHealth 0x800f0915` on 26200/26300.9278: https://www.reddit.com/r/Windows11/comments/1w0158t/august_27_2026kb5120998_os_builds_262009278_and/
- Microsoft — KB5120998 known issues (desktop settings/black background): https://support.microsoft.com/en-us/servicing/os/windows-11/2026/08/kb512099

Prepared from the live diagnostic session. Source logs use D/M/YYYY; times are local.