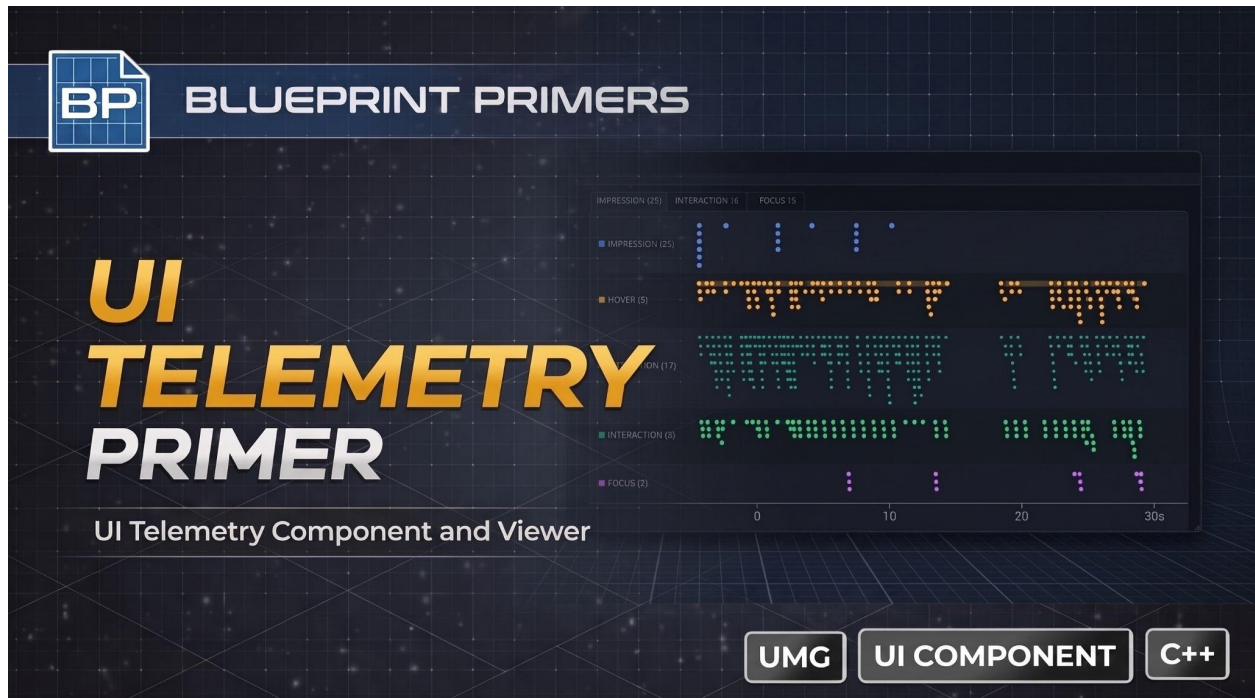




Primer UI Telemetry Component

Drop-on UMG telemetry — impressions, hovers, clicks, focus and value changes with zero graph edits

Attach a **UI Telemetry** component to any widget in the UMG Designer to instrument it, or drop a single **UI Telemetry Root** on a screen's top-level widget to cover its entire subtree. A central game-instance subsystem owns all policy — session identity, enrichment, consent, sampling, flood guarding, batching — and flushes event batches to pluggable sinks. The CSV sink writes analyzable session files by default.

Capture is client-side only and event-driven. Input is observed through one shared Slate preprocessor that never consumes events; impressions are measured from the widget paint pass (MRC-style viewability), not a per-frame poll.

How To

1. Instrument a Widget

Select any widget in the UMG Designer and **Add Component** → **UI Telemetry**. Set an **Event Id** (the identity its rows carry), optional **Static Tags**, and the track toggles: Impressions, Hover, Interactions, Focus, Lifetime.

2. Or Cover a Whole Screen

Select the screen's top-level widget (root Canvas / Overlay / Border) and **Add Component** → **UI Telemetry Root**. Every interactable beneath it is captured — a click on a button's label bills to the button, and rows a ListView pools into existence at runtime are captured like any other widget. Enable Skip Instrumented Widgets to let per-widget components take precedence without double counting.

3. Tune Viewability

Viewable Area Fraction and Viewable Time Threshold define when an impression counts (e.g. 50% visible for 1.0s). Project-wide policy — sampling chance, viewability engine, batching, heatmap capture — lives under **Project Settings > Primers > Primer UI Telemetry**.

4. Emit Your Own Events

Get Primer UI Telemetry Component (Widget) → Emit Custom Event (Extra Tags)

UI Telemetry Subsystem → Emit Custom Event (EventId, Tags) ← widget-less funnel markers

UI Telemetry Subsystem → Push Screen Context ("MainMenu") ... Pop Screen Context

Every event stamps the top of the screen-context stack; popping flushes the batch so a screen's events land together.

5. Play, Then Review

Play in PIE, then open **Tools > Primers > Telemetry Viewer** to explore the captured session. The default CSV sink's files load straight into the viewer or a spreadsheet.

6. Control Capture at Runtime

Every capture flag has a Blueprint getter and setter — toggle tracking, override sampling, or retag a widget live. Set Consent on the subsystem gates the whole pipeline for player-facing opt-outs.

7. Forward to Your Backend

Implement the four-method C++ sink interface and register it — batches arrive off the game thread, fire-and-forget. The bundled CSV, JSON, File and Log sinks are reference implementations.

Exposures

Component (UI Telemetry / UI Telemetry Root)

Function	Notes
Get Primer UI Telemetry Component (Widget)	Fetch the component from any widget.
Get Primer UI Telemetry Root Component (Widget)	Fetch the root component.
Get Owning Widget	The widget this component instruments.
Get Resolved Event Id	The effective identity after defaults.
Get / Set Event Id, Get / Set Static Tags	Retag at runtime.
Get / Set Track Impressions / Hover / Interactions / Focus / Lifetime	Per-signal toggles.
Get / Set Viewable Area Fraction, Get / Set Viewable Time Threshold	Impression rule.
Get / Set Sampling Override	Per-widget sampling; -1 inherits project policy.
Get / Set Log Telemetry	Per-widget debug logging.
Emit Custom Event (Extra Tags)	One-off event attributed to this widget.
Emit Telemetry Event (Event Type, Extra Tags, Duration Seconds)	Emit any event type manually.

Subsystem (UI Telemetry Subsystem)

Function	Notes
Emit Custom Event (EventId, Tags)	Widget-less event — funnel markers, purchases.
Push Screen Context (ScreenContext)	Push a logical screen name onto the stack.
Pop Screen Context	Pop and flush, so a screen's events land together.
Set Consent	Player-facing consent / kill-switch gate.

Delegates

Delegate	Fires On	Notes
OnPrimerRootHitCaptured (Component, HitWidget, EventType)	Root capture	After a root submits an event for a hit-tested widget; one handler serves any number of roots.

Data Types & Settings

Type / Setting	Notes
IPrimerUITelemetrySink	C++ sink contract: OnSessionBegin / WriteBatch / Flush / OnSessionEnd. WriteBatch may run off the game thread.
CSV / File / JSON / Log sinks	Bundled implementations; CSV on by default.
FPrimerUITelemetryEvent	Enriched event row.
FPrimerUITelemetrySessionHeader	Session identity snapshot handed to sinks.
UPrimerUITelemetrySettings	Developer settings: sampling, viewability, batching, heatmap.
WITH_PRIMER_UI_HEATMAP	Compile gate; heatmap capture is compiled out of Shipping unless defined =1 in target rules.

Notes

- Client-side only — telemetry has no authority model.
- A failing sink logs and drops; it never back-pressures the game thread.
- Batches flush event-driven: size threshold, screen-context pop, app deactivate, map load, session end.
- Session sampling is a per-session coin flip — a recorded session captures everything, so its data stays complete.
- With viewability disabled there is no per-frame sampling and no paint-observing wrapper at all.
- Demo content: Level_Demo_UITelemetry with W_Demo_Telemetry_PerComponent and W_Demo_Telemetry_FromRoot under the plugin's Demo folder.