

User Interface - Bug #11738

Fix FWD client memory leaks

08/13/2026 01:35 AM - Teodor Gorghe

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		reviewer:	
billable:	No	production:	No
vendor_id:	GCD	env_name:	
case_num:		topics:	
version_reported:			
version_resolved:			
Description			

History

#1 - 08/13/2026 02:04 AM - Teodor Gorghe

While [#11326](#) and [#11327](#) has decreased the minimal client memory requirement and memory spikes, there are some scenarios which causes memory leaks.
Constantin has made a memory leak fix in #11580 regarding a scenario which uses persistent triggers, but there are still some memory leaks for the FWD client.

When running the FWD client with **64M** heap, for a large GUI app, with intensive work, that session survives just a short time.

#4 - 08/13/2026 02:33 AM - Constantin Asofiei

What exactly is leaking?

#5 - 08/13/2026 02:35 AM - Teodor Gorghe

I don't know exactly what is leaking, but I have saw a substantial memory difference (+20M) cold vs hot (on the same initial screen).
Capturing the heap dumps right now.

#6 - 08/13/2026 02:42 AM - Teodor Gorghe

Attached on devsrv01:/tmp/tg.20260813/ (cold.hprof and hot.hprof).

Analyzing right now.

#7 - 08/13/2026 02:44 AM - Constantin Asofiei

Teodor Gorghe wrote:

Attached on devsrv01:/tmp/tg.20260813/ (cold.hprof and hot.hprof).

Analyzing right now.

There is no cold.hprof.

#8 - 08/13/2026 02:45 AM - Teodor Gorghe

Done.

#9 - 08/13/2026 03:21 AM - Teodor Gorghe

This is what AI has found during heap dump analysis. I will do the same smoke test and also record the heap dump on the server-side.

Heap leak analysis - cold vs hot

Both dumps: same screen, idle, after GC, 10 minutes apart. Live heap grew 43.1 MB -> 51.9 MB (+8.8 MB, ~53 MB/hour). About 6.0 MB sits in the maps below.

Leaks found

No	Container	Grew	What it pins
1	FontManager\$WorkArea.orphanFontResolvers	32 -> 1458	1387 dead radio buttons, 2.92 MB
2	AbstractGuiDriver.pendingTtWorkers	86 -> 455	138 browse columns, 66 dead menus
3	WidgetRegistry.widgetList	423 -> 806	widgets never unregistered
4	ClientConfigManager.widgetCfg / activeConfigs / backupConfigs	391 -> 770 each	button, browse and frame configs
5	AbstractGuiDriver.widgetsRegions	164 -> 1540 rects	bounds cache, 29 window ids from 3
6	MouseHandler\$WorkArea.actions / cachedActions	105 -> 157, 155 -> 194	mouse action lists
7	GuiWebDriver.imageUsages / imageSeals	42 -> 375 each	image bitmaps

Root cause

All seven are widget-keyed side tables. Each is cleaned only on widget teardown, and teardown never runs.

- RadioSet.refreshItems() calls buttons.clear() and rebuilds every button. The old buttons are never destroyed. There is no destroy call anywhere in RadioSet.java, RadioSetGuiImpl.java or RadioButtonGuiImpl.java.
- Browse columns and menus are dropped the same way.
- Because destroy() never runs, notifyWidgetDestroyed(), WidgetRegistry.remove() and the MouseHandler cleanup never run either.
- The strong map keys then pin each dead widget plus its resolvers, configs, insets, points and labels.

Not leaks

- AbstractGuiDriver.textMetricsCache - 341 -> 1995, but it is an LFUAgingCache that evicts when full. Cache warm-up, ~0.55 MB.
- MethodHandle, MemberName, MethodType, Method, Constructor growth - JVM lambda and reflection warm-up, ~0.35 MB.

Still open

About 1.9 MB of widget-tree growth: FrameGuiImpl 33 -> 55, ScrollPaneGuiImpl 33 -> 58, BorderedPanelGuiImpl 72 -> 143, ScrollBarGuiImpl 59 -> 121, ButtonConfig 282 -> 582. Dead subtrees still hanging off live parents. Same root cause, but no single retaining edge isolated yet.

#10 - 08/14/2026 02:12 PM - Teodor Gorghe

Made some improvements regarding fixing some FWD client memory leaks.

Initially, the heap was increasing with **12M** on that 5 minute smote test.

I have fixed the radio button memory leak and the one related with the widgetList, and the leak has reduced to **6M**.

#11 - 08/27/2026 03:27 AM - Teodor Gorghe

Created task branch **11738a** and committed revision **16725**:

- Added initial work: Fixed widget and radio set memory leaks.