

## User Interface - Support #4059

### UI performance improvements (master task)

05/01/2019 11:18 AM - Greg Shah

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #1 - 05/01/2019 11:25 AM - Greg Shah

This task is a master index of planned/proposed UI performance work:

- [#4912](#) move UI portions of the web client to the server-side
- [#4029](#) implement more widget rendering knowledge within the client drivers
- [#4955](#) web client/javascript drawing - fast vs crisp
- [#2672](#) GUI web client performance improvements (also is related to [#2913](#) window dragging)
- [#2613](#) improve performance of ScreenBitmap.canDraw (and of its callers)
- [#3128](#) header expressions (and dynamic row/column/title...) may be executing more than is necessary
- editor/fillin editing is very slow
- enable/disable OS events is very expensive and is being done more than necessary
- disable logging by default in GUI (Java and JavaScript)
- [#4026](#) ensure all objects which are transmitted over the DAP implement Externalizable
- [#4028](#) coalesce STATUS updates sent in rapid succession, such as for a running progress indicator
- [#4062](#) LT.getChanges() improvements
- [#3246](#) reduce the amount of data being sent to the client-side when an UI attribute is being changed
- [#4134](#) reduce the processing for calculated columns
- [#2244](#) ChUI web client performance
- [#5582](#) webgl renderer is significantly slower than canvas2d (performance regression)
- [#5583](#) MenuGuiImpl redraws everything when only a subset of drawing is needed
- [#5584](#) WidgetRegistry.getComponent() is expensive and can be optimized