

# An Evaluation of EDM Replacement Candidates at SLAC

Alisha Babbitt, Lucas Jose Monteiro Carvalho, Matthew Gibbs,\* and Ernest Williams, Jr.  
*Stanford Linear Accelerator Center*  
(Dated: September 15, 2015)

SLAC has historically used EDM (Extensible Display Manager) as its EPICS graphical user interface technology. EDM is based on the Motif widget toolkit, which is nearing end-of-life, and therefore EDM must be replaced. We discuss the requirements of a display manager at SLAC and evaluate the performance and feature set of several replacement candidates against those requirements. For each candidate, we discuss additional work that would be required to match EDM's features. Finally, we discuss additional features not present in EDM that we would like to see in a replacement.

## I. DISPLAY MANAGER REQUIREMENTS AT SLAC

For many years, SLAC has used EDM [1] as the primary graphical user interface for both the Linac Coherent Light Source (LCLS) and the SPEAR3 storage ring control systems. While SLAC is for the most part quite happy with EDM, it is starting to show its age in a few ways. Its reliance on the Motif widget toolkit is a liability, as Motif is nearing end-of-life, and may not be usable in the future. The Motif-based interface is rather dated, and it does not visually fit in well with other applications using more modern GUI toolkits like Qt or GTK+. EDM does not lend itself to high-quality text rendering, often leading to either large ugly pixellated text, or tiny, hard-to-read text. A modern user interface would not only solve the Motif end-of-life issue, but also provide a better experience.

SLAC uses almost every feature available in EDM, and nearly all of the available EDM widgets appear in at least one panel. In addition, SLAC has many critical displays that rely on EDM's fast drawing performance - displays with several hundred widgets that need to update at roughly 30 Hz. Operators running the accelerator often have 10-30 displays open at the same time, and need to be able to do this without using up all the available resources of their workstations. Any suitable replacement must be able to meet the same demands we have for EDM in features and performance.

Creating EDM panels is done within the EDM program itself, in an editing mode. The editing mode uses a drag and drop, WYSIWYG interface that makes it quite easy for anybody to make a basic panel, even beginners. SLAC operators, even those who have little or no programming experience, make custom panels to make operations tasks easier. This has proven to be very beneficial, and lets operators run the machine very efficiently using tools of their own design. A good EDM replacement needs to make it very easy to author basic displays.

SLAC has many basic displays, but there are also some which are very complicated, with many nested displays, local PVs, and macros. EDM display authors at SLAC frequently run into limitations when trying to create com-

plex, highly interactive displays that use some of the more complex conventions of modern graphical user interfaces. These limitations sometimes stem from EDM's widget set, and other times stem from the difficulty in adding client-side application logic to a display. Even something as simple as a pop-up dialog box with a text field can be tricky to implement in EDM. A strong replacement candidate must match EDM's existing widgets and features, but SLAC's panel authors and operators also demand that a next-generation user interface do more to enable the creation of powerful, high-level tools for running the accelerator.

## II. REPLACEMENT CANDIDATES

There are three EPICS display manager projects which we considered in our tests: EpicsQt[2], caQtDM[3], and CSS-BOY[4]. Our initial goal was to evaluate Qt-based display manager options, and EpicsQt and caQtDM seem to be the two most mature options, with the largest amount of EPICS community support. We also compare against CSS-BOY, a component in the very popular Control System Studio package. CSS-BOY is based on the Eclipse Rich Client Platform and Standard Widget Toolkit for Java.

All three options have easy to use WYSIWYG editors available for panel design. EpicsQt and caQtDM both use Qt Designer, which is part of the standard Qt tool set. CSS-BOY uses its own editor. Both editors feature palettes of widgets that can be dragged onto a blank window.

### A. EpicsQt

EpicsQt is a Qt framework developed at the Australian Synchrotron. EpicsQt was designed as a successor to both EDM and MEDM, but not necessarily a direct replacement to either. As such, it does not try to completely duplicate the EDM feature set or widget set. EpicsQt provides an EPICS aware widget set which can

be used in simple, code-free, EDM-style displays, or in custom C++ applications.

## B. caQtDM

caQtDM is a Qt framework developed at the Paul Scherrer Institute (PSI). caQtDM was explicitly developed as a replacement for MEDM [5]. MEDM is another display manager, similar to EDM in many ways, but does not have all of EDM’s features, so caQtDM does not have all of EDM’s features either. caQtDM’s architecture consists of widgets with no EPICS capability, driven by a main program which performs all the EPICS operations.

## C. CSS-BOY

CSS-BOY is part of the Control System Studio (CSS) package, which is a collection of control system tools based on the Eclipse Rich Client Platform. CSS is developed by a collaboration of many large accelerator facilities: DESY, SNS, BNL, FRIB, and ITER. It has strong support in the EPICS community.

# III. EVALUATION OF REPLACEMENT CANDIDATES

## A. Widget Set Evaluation

EDM has a widget set that is well-suited for running a typical accelerator. It provides widgets for visualizing and manipulating analog values using graphical meters and sliders in addition to standard user interface elements like buttons, text fields, and labels. It has widgets for plotting signals vs. time, plotting waveform signals, and displaying images from cameras and other 2D sensor arrays. EDM also includes a few widgets that work as user interface controls - the most notable is the “menu mux”, which can change custom variables (“macros” in EDM parlance) used by other widgets on the display.

For each of the 47 EDM widgets, we found the closest counterpart in the widget sets for the candidate programs, if a counterpart existed at all. For each candidate, a table was created listing the EDM widget, its counterpart widget, and any notes about differences in widget behavior. For each EDM widget, a rating is assigned: “green” if a counterpart exists with only minor cosmetic differences or no differences at all, “yellow” if the counterpart has cosmetic or behavioral differences, and “red” if there is no clear counterpart, if the counterpart is missing critical functionality, or if the widget has bugs that make it unfit for use.

Two of the display manager candidates, caQtDM and CSS, come with automatic EDM conversion scripts,

which take EDM files as input, and convert them to the new display manager’s format. These conversion scripts can alleviate some (but not all) of the repetitive work involved in converting the large number of existing EDM displays. Of course, these scripts can only convert EDM widgets that have a counterpart in the new display manager, and all the features of the EDM widget might not be implemented in the new display manager.

Under our rating system, it should be possible to use an automatic conversion script to reliably convert an EDM display with only “green” widgets, with only cosmetic changes needed to make the new display usable. An EDM display with “yellow” widgets might need some manual re-design after conversion to make it operable, or alternatively, the widget could be improved to get it into a “green” state. An EDM display that uses “red” widgets would not be possible to make, without adding functionality to existing widgets, or creating a new widget entirely.

Thirteen of EDM’s widgets are unused at SLAC, or are used very rarely. For these widgets, we note the closest counterpart, but do not rate the widget. Note that this leaves 34 widgets that need a counterpart.

Brief summaries are below. The full tables are available in appendixes A, B, and C.

### 1. *EpicsQt Widget Set*

In EpicsQt’s widget set, 19 EDM widgets have a “green” counterpart, 8 EDM widgets have a “yellow” counterpart, and 7 EDM widgets are “red”, and have no counterpart, or an incomplete counterpart. In our opinion, the three most significant EDM widgets without a good EpicsQt counterpart are the Menu Mux widget, the Symbol widget, and the X-Y Graph. Menu Mux and Symbol have no EpicsQt counterparts at all. The X-Y Graph has two partial EpicsQt counterparts: “QEPlot” and “QEPlotter”, but neither of them support all the features of X-Y Graph which are used at SLAC.

### 2. *caQtDM Widget Set*

caQtDM’s widget set is very similar to EDM’s. 23 EDM widgets have a “green” caQtDM counterpart, 9 EDM widgets have a “yellow” caQtDM counterpart, and 2 EDM widgets are “red”. The two “red” EDM widgets are the Symbol widget, and Menu Mux widget.

### 3. *CSS-BOY Widget Set*

CSS-BOY’s widget is also a close match to EDM. 23 EDM widgets have a “green” CSS counterpart, 9 widgets have a “yellow” counterpart, and two widgets are “red”.

The two “red” widgets are the Menu Mux widget, and Coef Table.

## B. Performance Evaluation

We used two quantitative metrics to evaluate the performance of the display manager candidates: memory usage and CPU usage. We also discuss the results of a performance study done by Farnsworth et al [6] at Argonne National Laboratory, which measured the graphical update rates for the candidates.

### 1. Multiple Display Test

Our test measures the display manager resource usage as a function of the number of displays open. For each display manager, one instance was opened, then one display opened. After collecting an average of CPU and memory usage, another copy of the display is opened, and the measurement is repeated. We continue this process, incrementing the number of displays until 100 displays are on screen simultaneously. For reference, LCLS operators often have somewhere between 10 and 30 panels open simultaneously. Figure 1 shows the results for CPU usage, and Figure 2 shows the results for memory usage. EDM uses very little memory and CPU, even with large numbers of displays open. caQtDM and epicsQt both show reasonable CPU usage and memory usage, certainly within acceptable bounds for modern workstations. CSS uses a very large amount of CPU (even with one screen CPU usage is very high). CSS memory usage is also much higher than the other options (although with tens of gigabytes of RAM being common, CSS is still not close to the limits of system memory).

### 2. Graphics Update Rate

A group at the APS at Argonne National Laboratory investigated several display managers as potential MEDM replacements. As part of their study, they created panels with hundreds of widgets connected to PVs that updated at 10 Hz. They recorded the number of widgets they could display at the same time without skipping any of the 10 Hz PV updates (the lossless widget number), as well as the number of widgets that could be displayed before the user interface completely failed (the max widget number). The update rate data is given in Table I.

The results from APS suggest that all of the options are viable even for displays with hundreds of frequently updating widgets. Our own experience doesn’t completely agree with that. For each candidate, we created an LCLS “orbit display”, modeled after a very heavily used EDM

display at LCLS. The orbit display represents the X, Y, and intensity signals from every LCLS Beam Position Monitor as a set of three bar widgets. There are 176 BPMs displayed, for a total of 528 bar widgets on a single panel. These bar widgets are connected to PVs that update at 10 Hz.

We found that the LCLS orbit display had very poor performance in both caQtDM and EpicsQt, with screen update rates clearly less than 10 Hz. CSS-BOY’s performance with the orbit display was noticeably better than the Qt-based options, but worse than that of EDM. Unfortunately, we did not devise a good way to quantitatively measure the update rate.

During our exploration of graphics performance and Qt, we created our own LCLS orbit display in PyQt [7] (a set of Qt bindings for the Python programming language), with performance similar to that of EDM. It is our conclusion that using Qt does not rule out the possibility of high performance graphics, but developers need to be careful if they want to achieve it.

## IV. WORK NEEDED TO MATCH EDM

There is no one-to-one ‘drop-in’ replacement for EDM - every option would require some effort to modify converted displays to work in the new display manager, or modify the display manager to more-closely reproduce the behavior of EDM, or (most likely) both.

SLAC’s existing 4688 EDM display files would need to be converted to the new display manager. In our experience, recreating a fairly complicated EDM display (with 10 different kinds of widgets, and about 50 total widgets) took about an hour. CSS and caQtDM both include conversion scripts to automate the conversion from

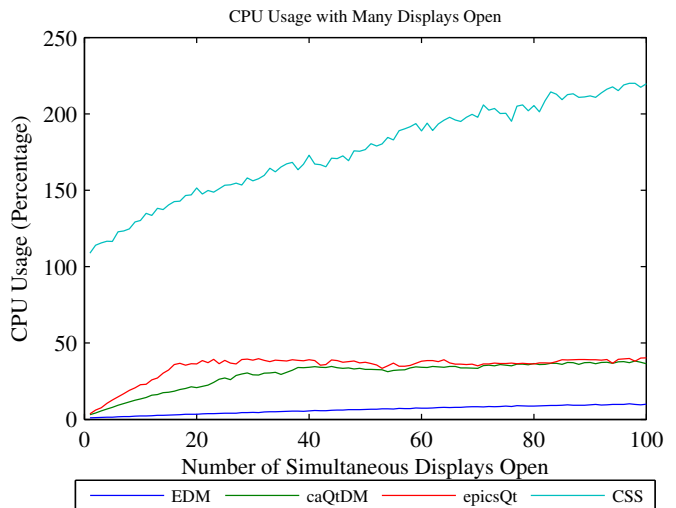


FIG. 1. CPU Usage for each of the candidates, as well as EDM, as a function of simultaneous number of displays open.

EDM. We found that using these scripts to convert a display produced mostly-functional results, but still needed nearly an hour of work to test the converted display, and clean up graphics issues. Some of this work could be eliminated with improvements to the conversion scripts. For epicsQt, a completely new conversion script would have to be built. Even with a conversion script, the process cannot be fully automated, as each converted display needs to be tested to ensure functionality.

Automatic conversion of a display to a new display manager will fail if there is not an equivalent widget for every EDM widget. As discussed above, there are missing widgets for every candidate. There are two options for dealing with this problem: re-design all panels that use the widget, or build a replacement widget. For rarely-used widgets, it may be preferable to re-design the panel. If the EDM widget is frequently used, it makes more sense to build a replacement widget for it.

For the two Qt-based options, we investigated the work required to create a replacement widget. Qt widgets are written in C++. For both epicsQt and caQtDM, the average widget is about 300 lines of code, and complicated widgets are closer to 1200 lines. For caQtDM, adding a new widget also requires modifications in caQtDMLib, which handles EPICS communication for the widgets. The work required to create a new widget in CSS-BOY was not investigated. As discussed in Section III A, the number of missing widgets vary among the candidates, with epicsQt having the most widgets needing replacement, and caQtDM and CSS-BOY tied for fewest number of widgets needing replacement.

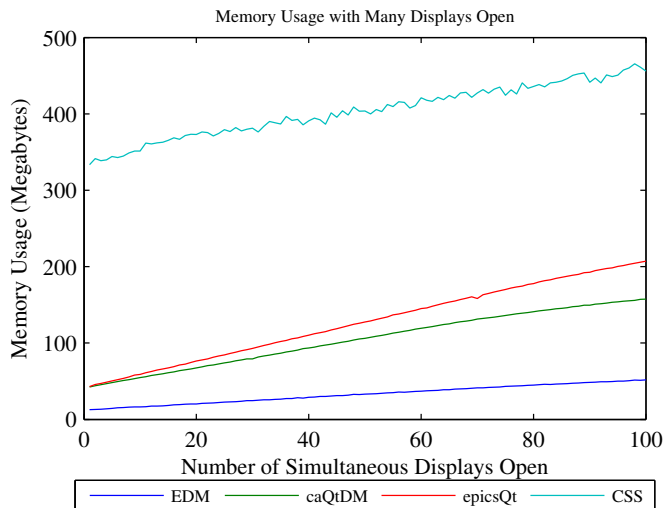


FIG. 2. Memory Usage for each of the candidates, as well as EDM, as a function of simultaneous number of displays open.

## V. ADDITIONAL FEATURES BEYOND EDM

Finding a replacement that matches EDM’s feature set is important, but simply duplicating EDM would be a missed opportunity to improve upon its shortcomings. While gathering requirements from the many groups at SLAC that use EDM, we were repeatedly presented with areas where users felt EDM needed improvement.

### A. Window Management

A common problem with EDM at SLAC is that most tasks require operators to go several levels deep into a hierarchy of displays, and each new level opens a new window. After running the machine for a while, very large numbers of open displays accumulate unless the operator is diligent about closing windows. A more elegant window management system would be beneficial to users. caQtDM, being a clone of MEDM’s functionality, has no new window management features. EpicsQt and CSS-BOY both offer a tabbed-window system, that operates similar to most modern web browsers, which can significantly cut down on the number of windows open at once. EpicsQt’s implementation feels like a bit of an afterthought, and has sometimes unpredictable behavior: related display buttons configured to open new displays as tabs often open in a new window instead. CSS-BOY has better tab support, but weaker support for multiple windows open at the same time, which is still useful in many instances.

Another way to cut down on the window management burden would be to make navigation of the control system more efficient. Web browsers have a very good model: In addition to links between pages, a navigation stack lets you move backward and forward between recently opened pages, and an address bar to let you go directly to a particular page. The ubiquity of the web means that nearly everybody is already familiar with this model. Unfortunately, none of the replacement candidates we studied use a system like this.

TABLE I. Number of widgets displayed on a panel before the panel skips 10 Hz updates (lossless count), and number of widgets displayed before the panel is non-operational (max count), as measured by Farnsworth et al.

Display Manager	Lossless Widget Count	Max Widget Count
EDM	3200	14288
caQtDM	1200	1660
EpicsQt	400	1440
CSS-BOY	800	984

## B. Support for Building Complex Applications

EDM displays are very simple: they are a collection of widgets which redraw themselves when a PV changes. This simplicity makes EDM very easy to learn, but also makes it unsuitable for building applications that require more client-side computations, or more business logic. While most of the time, business logic belongs in the IOC, there are legitimate instances where client-side computation is the right choice. In these cases, SLAC has typically abandoned EDM for other, more general purpose development frameworks, like MATLAB, Java, or Python, where building EPICS functionality into widgets is time-consuming and potentially buggy. A SLAC-wide standard framework for building high-level applications with a mix of EPICS-aware widgets and custom code could save significant development time, and give users a way to make the kinds of applications that make it easier to perform complex operating or analysis tasks.

EpicsQt and caQtDM can both be embedded into custom C++ applications, but neither are particularly easy, and are out of reach for users with limited programming experience. CSS-BOY includes a javascript-based scripting system, where each widget has an embedded script which can contain business logic. These scripts allow for great flexibility, but introduce performance problems as the number and complexity of scripts increases. CSS's developers discourage the use of all but the simplest scripts.

Several groups approached us with the suggestion that they would like to be able to use the EPICS-aware widget set of these frameworks inside PyQt, which has a large SLAC user base. This is only possible with the two Qt-based options. PyQt has a tool called "SIP" [8] which can be used to create Python bindings for widgets written in C++ (like those of caQtDM and EpicsQt). SIP has a steep learning curve, but it should be possible to create bindings for the caQtDM and EpicsQt widget sets. While learning the procedure, we created a Python binding for a very simple C++ widget in about two days. Due to the way caQtDM is implemented, using the widgets in PyQt applications would take more work than simply creating bindings, because caQtDM's main program handles all the EPICS communication - not the widgets. An alternative way to establish EPICS communication and update the widgets would need to be built into the PyQt application.

## VI. CONCLUSIONS

Our study of the existing Qt-based display managers leads us to conclude that Qt is a good foundation for a display manager. Qt itself is powerful and performant, straightforward to use, and has a strong support community. However, we find that in their current state, none

of the existing Qt-based options meet all of the requirements our users demand.

### A. EpicsQt

Of the three options, EpicsQt would take the most work to replicate the EDM widgets we need. EpicsQt's performance is adequate for most tasks, but it struggles when drawing hundreds of rapidly-updating widgets. EpicsQt has a feature for organizing displays as tabs, which could make window management easier, but this feature did not work as expected. The ability to create complex applications with client-side logic is possible in EpicsQt, by using their "QCa" library and the EpicsQt widgets in a C++ application, but this is not a very attractive option for most of our users, who want the ease and flexibility of a scripting language.

### B. caQtDM

caQtDM comes quite close to duplicating EDM's functionality and widget set, with only two EDM widgets that would need to be recreated. caQtDM was the faster of the two Qt based solutions, and has acceptable memory and CPU usage even under high load. Much like EpicsQt, it had problems with handling hundreds of quick widget updates. Unfortunately caQtDM is lacking many of the new features our users require: it offers no improvements to window management over EDM, and making a full application again requires including the caQtDM widgets and the caQtDM library in a C++ application.

### C. CSS-BOY

CSS-BOY is an alternative to a Qt-based display manager. Its widget set is very similar to EDM's (our analysis rates it the same as caQtDM). CSS-BOY had the highest memory and CPU usage of the three candidates by a considerable margin, but was also the fastest at updating very large numbers of widgets. Start-up time was much slower in CSS-BOY than in the other options. CSS-BOY has a tabbed window system, which helps cut down on window clutter. CSS-BOY also has scripting support, but these scripts are somewhat limited in their capabilities, and are a frequent source of performance problems. It would be difficult to create scripted CSS-BOY displays to replace existing MATLAB or PyQt applications used at LCLS.

---

\* E-Mail Address: mgibbs@slac.stanford.edu

- [1] J. Sinclair, “Extensible Display Manager,” <http://ics-web.sns.ornl.gov/edm/>, [Online; accessed 10-August-2015].
- [2] G. Jackson, A. Owen, A. Rhyder, A. Starritt, and R. Fernandes, “EPICS Qt Framework,” <http://sourceforge.net/projects/epicsqt/>, [Online; accessed 11-August-2015].
- [3] A. Mezger and H. Brands, “caQtDM - a medm replacement based on QT,” <http://epics.web.psi.ch/software/caqtdm/>, [Online; accessed 11-August-2015].
- [4] C. Collaboration, “CSS - Control System Studio,” <http://controlsystemstudio.org/index.html>, [Online; accessed 11-August-2015].
- [5] A. Mezger and H. Brands, in *Proceedings of ICALEPCS2013*, ICALEPCS (San Francisco, 2013) pp. 864–866.
- [6] R. Farnsworth, J. P. Hammonds, B. Pausma, C. Suarez, A. Rhyder, and A. C. Starritt, in *Proceedings of ICALEPCS2013*, ICALEPCS (San Francisco, 2013) pp. 547–549.
- [7] Riverbank Software, “What is PyQt?” <https://www.riverbankcomputing.com/software/pyqt/intro> (), [Online, accessed 12-August-2015].
- [8] Riverbank Software, “What is SIP?” <https://www.riverbankcomputing.com/software/sip/intro> (), [Online, accessed 12-August-2015].

## Appendix A: EpicsQt Widget Analysis

EDM Widget	EpicsQt Counterpart	Notes
Lines	QEShape	Limited to 10 points in QEShape.
Rectangle	QESimpleShape	
Circle	QESimpleShape	
Arc	QEShape	
Static Text	QESubstitutedLabel	
Text w. Reg. Exp.	None	Text w. Reg. Exp. is unused at SLAC.
Embedded Window	QEForm	
PNG Image	QEFileImage	QEFileImage takes a filename from a PV. You cannot specify a plain string as the filename.
GIF Image	QEFileImage	QEFileImage takes a filename from a PV. You cannot specify a plain string as the filename.
Dynamic Symbol	None	No epicsQt counterpart, but this widget is unused, at least in LCLS EDM panels.
Meter	QEAnalogProgressBar	Set the 'mode' property to 'meter.'
Bar	QEAnalogProgressBar	Even with alarm-sensitive mode off, always resets the background color to some default, regardless of what you choose. You cannot have a bar with an origin not at the top or bottom.
Message Box	None	The EDM Message Box widget has no epicsQt counterpart. This widget is only used in one LCLS EDM Panel, which was used for vacuum commissioning purposes. Not essential for SLAC's purposes.
X-Y Graph	QEPlotter or QEPlot	QEPlotter for waveforms, QEPlot for striptools or waveforms. There is no mode to do a cartesian plot (where one variable controls X for a single data point and another controls Y). QEPlot breaks if you try to mix waveform PVs and Scalar PVs. There is no way to set line thickness on QEPlot or QEPlotter. QEPlotter only supports auto-scaling right now, which is annoying. The scales keep changing, so the plot sort of jumps around. If you screw up and put a bad value into X and A, it can cause designer to lock up.
Text Monitor	QELabel	No 'Clip to LOPR/HOPR' feature. QELabel refreshes very quickly, which can make it hard to read for PVs that update very rapidly. EDM Text Monitor seems to do some kind of rate limiting on its updates.
PV Inspector	QEPVProperties	Unused at SLAC.
Table	None	The EDM Table widget uses a PV as a filename, and tries to open a text file, and display the contents of that file as a table. There is no epicsQt replacement. This widget is not used in very many panels, but the panels that have it are used quite frequently (BCS/PPS CUD is a big one).
Coef Table	QETable	QETable cannot show titles for each row (which corresponds to an element in the array PV).
Indicator	QEAnalogProgressBar	Set the 'mode' property to 'Scale.'
Text Update	QELabel	Alarm coloring changes the background on QELabel, but changes the text color on Text Update. Cannot choose a separate PV for alarm coloring.
Reg Text Update	None	The Reg Text Update widget lets you define a regular expression to show or hide the text. While this widget is used frequently, the regex feature of it is not used in any LCLS panel, so a conversion to QELabel would work fine.
Byte	QEBitStatus	
Hoff Video	QEImage	

Multi-line Text Update	None	Multi-line text update is a text update that can display multiple lines of text, in the case that it is displaying a long string PV with line breaks. There is no direct epicsQt counterpart, but the multi-line feature of the EDM widget is not used in any LCLS EDM panels, so it can be safely replaced with QELabel.
SLAC XY Graph	QEPlotter or QEPlot	SLAC XY Graph is functionally the same as the X-Y Graph widget, with one difference: if there is a mix of valid and invalid PVs specified for the traces, the valid ones will still plot. QEPlot works the same way - invalid PVs don't stop valid ones from working. See notes for X-Y Graph.
Older optimized version of Bar	QEAnalogProgressBar	The 'Older optimized version of Bar' widget is an older version of the widget that had extremely fast drawing performance. It is used in some LCLS EDM panels to show very quickly updating values, like 10 Hz data from BPMs. The QEAnalogProgressBar widget is much slower to draw than this fast bar widget.
Symbol	None	The Symbol class lets you define an array of groups of graphics primitives, and display a different item from that array based on the value of a PV. It is used in many LCLS panels, to graphically represent things like a PPS door. There is no straightforward epicsQt counterpart.
Animated Symbol	None	An animated version of the Symbol class, where there are additional PVs that control the position and rotation of the symbol. There is no epicsQt counterpart for this widget, but it is unused in any LCLS EDM Panels.
Text Control	QELineEdit	The Text Control widget lets you click on the text to edit it. A pop-up dialog box is displayed that lets you change the value of the PV. epicsQt doesn't have a widget that duplicates this exact behavior, but it does have QELineEdit, which is a standard text entry field.
Text Entry	QELineEdit	
Slider	QESlider	QESlider has the save/restore buttons like the EDM version. Graphically, it looks very different, which may make converted panels look strange.
Motif Slider	QESlider	Doesn't look too much like the Motif Slider, but works similarly. Can't completely disable the scale. There is no way to set the click increment value.
Button	QECheckBox or QEPushButton	The EDM Button widget has two modes: Push and Toggle. To duplicate a Push mode button use QEPushButton. To duplicate a Toggle mode button use QECheckBox.
Menu Button	QEComboBox	
Radio Box	QERadioGroup	
Message Button	QECheckBox or QEPushButton	EDM's Message Button is like the 'Button' widget, but it has a few additional features. These don't exist in epicsQt, like 'Close Display on Press'. More investigation is needed to see if we need these features.
Up/Down Button	None	EDM's Up/Down button lets you increment or decrement the value of a PV with left and right mouse buttons. It is only used in four LCLS panels. This button is extremely confusing to use, and probably shouldn't be duplicated.
Ramp Button	None	EDM's Ramp button continuously changes the value of a PV from a minimum value to a maximum. There is no epicsQt counterpart. This button is only used in one LCLS EDM Panel. The ramp button functionality could be duplicated with a shell script plus a QEPushButton.
Freeze Button	None	The Freeze Button freezes all activity on an EDM panel. There is no epicsQt counterpart. This button is rarely used in LCLS panels, and is usually just a convenience to make it easier to take a screenshot of the panel.



Exit Button	QPushButton	The EDM Exit Button closes the current panel. It also has an option to close the parent panel, if it is used in an embedded window, and can also exit the whole EDM program. It is used on a huge number of LCLS panels, but the functionality is really more of a minor convenience than an essential feature. The user can just close the window with the standard window ‘close’ button (the ‘X’ button at the top of the window), or use a QPushButton, and connect the clicked() signal to the parent widget’s close() slot.
Menu Mux	None	Menu Mux is used in many panels, and there is no direct replacement for it. Menu Mux lets you create a drop-down menu with multiple choices. Each choice sets values for multiple macro variables.
Related Display	QEPushButton	Use the ‘guiFile’ field. The ‘creationOption’ property might be very useful (lets you open displays in tabs and docks), but it didn’t work properly in testing.
Shell Command	QEPushButton	Use the ‘Program’ and ‘Arguments’ fields.
Choice Button	QERadioGroup	There is no epicsQt widget that uses the same visual style as the Choice Button, but QERadioGroup duplicates the functionality.
Multi-line Text Entry	QELineEdit	The Multi-line Text Entry widget lets you change a multi-line text string PV. This widget is used in one LCLS EDM panel (a test panel for LI28 LLRF), and on that panel it could be replaced with a single line text entry. So, for our purposes, QELineEdit is a fine replacement option.
Triumpf Slider	QEAnalogSlider	Unclear what the difference is between Triumpf slider and Motif slider. QEAnalogSlider is a decent replacement for the Motif Slider (see above notes for Motif slider).
Multi-Slider	None	Works like the EDM Slider class, but has controls for two PVs on the same slider. Used on a couple of EDM panels for oscilloscope control. Could be replaced with two separate sliders.

TABLE II: Analysis of the EpicsQt widget set, as compared to EDM.

### Appendix B: caQtDM Widget Analysis

EDM Widget	caQtDM Counterpart	Notes
Lines	caPolyLine	In Qt Designer, you can’t directly draw the lines, you have to specify a set of point coordinates.
Rectangle	caGraphics	In the ‘form’ attribute of caGraphics, use the ‘Rectangle’ option.
Circle	caGraphics	In the ‘form’ attribute of caGraphics, use the ‘Circle’ option.
Arc	caGraphics	In the ‘form’ attribute of caGraphics, use the ‘Arc’ option.
Static Text	caLabel	
Text w. Reg. Exp.	caLabel	Text w. Reg. Exp. lets you show or hide a text label based on a regular expression evaluation of the label’s contents. This widget is used in a large number of LCLS EDM panels, but the regular expression feature is not used by any LCLS panels. In all SLAC use-cases, this could be replaced by caLabel.
Embedded Window	caInclude	The Embedded Window widget lets you embed one EDM file within another. You can specify the filename via a PV, hard-coded string, or it can be used with the EDM ‘Menu Mux’ widget. In caQtDM, caInclude does not let you specify the file via a PV, and it does not have a Menu Mux. Behavior very similar to ‘Menu Mux’ can be created with a QTabWidget which contains multiple caInclude widgets.
PNG Image	caImage	
GIF Image	caImage	
Dynamic Symbol	None	The Dynamic Symbol class lets you build simple animated graphics objects out of graphics primitives, like lines, rectangles, etc. This widget has no caQtDM counterpart, but it is not used in any LCLS EDM Panels, so this isn’t a big issue.

Meter	caMeter	caQtDM has two very similar widgets that both work like EDM's Meter widget. In the case of a 'Total Display Angle' of 180 degrees, which is commonly used in LCLS EDM Panels, the caMeter widget takes up more space, because it always draws a full circle. Converted displays that use this widget may need to be slightly redesigned.
Bar	caThermo	The EDM Bar widget has an 'origin' property that can be set to any value. caThermo only lets you set the origin to a few fixed values (bottom of bar, top of bar, center of bar), via the 'type' property. We could not find any good examples of any LCLS EDM panels that utilize this extra flexibility, but it is hard to determine for sure that it is completely unused. The conversion script probably won't know how to set the 'type' value properly.
Message Box	None	The EDM Message Box widget has no caQtDM counterpart. This widget is only used in one LCLS EDM Panel, which was used for vacuum commissioning purposes. Not essential for SLAC's purposes.
X-Y Graph	caCartesianPlot or caStripPlot	
Text Monitor	caLineEdit	Text Monitor has two features that caLineEdit doesn't duplicate: 'Clip to LOPR/HOPR', and 'Null PV'. 'Null PV' is effectively unused in LCLS panels (it is usually set to the same PV as the 'PV' property, or not set at all). 'Clip to LOPR/HOPR' is used in many panels, but it is not clear that it is essential. There is no ability to set a separate PV for alarm coloring.
PV Inspector	QEPVProperties	There is no caQtDM replacement for the PV Inspector widget, but PV Inspector is not used in any LCLS EDM panels.
Table	None	The EDM Table widget uses a PV as a filename, and tries to open a text file, and display the contents of that file as a table. There is no caQtDM replacement. This widget is not used in very many panels, but the panels that have it are used quite frequently (BCS/PPS CUD is a big one).
Coef Table	caWaveTable	The Coef Table shows the individual values of a waveform PV as rows in a table. The caQtDM counterpart caWaveTable defaults to showing each value in a new column, but by changing the number of columns and rows you can duplicate the EDM default. caWaveTable does not let you define a custom label for each element.
Indicator	caLinearGauge	caLinearGauges with the scale labels enabled look bad if you aren't careful to keep the right aspect ratio.
Text Update	caLineEdit	
Reg Text Update	None	The Reg Text Update widget lets you define a regular expression to show or hide the text. While this widget is used frequently, the regex feature of it is not used in any LCLS panel, so a conversion to caLineEdit would work fine.
Byte	caByte	
Hoff Video	caCamera	In the caCamera widget, the value of the 'channelCode' PV must be set to 1 for a grayscale camera. Strangely, the value of the 'channelBPP' PV must be set to 1 for an 8 bit image. If that is because 'BPP' is 'Bytes Per Pixel' instead of 'Bits Per Pixel', we might have problems with some cameras where the number of bits per pixel is not divisible by 8.
Multi-line Text Update	caLineEdit	Multi-line text update is a text update that can display multiple lines of text, in the case that it is displaying a long string PV with line breaks. There is no direct caQtDM counterpart, but the multi-line feature of the EDM widget is not used in any LCLS EDM panels, so it can be safely replaced with caLineEdit.
SLAC XY Graph	caCartesianPlot or caStripPlot	SLAC XY Graph is functionally the same as the X-Y Graph widget, with one difference: if there is a mix of valid and invalid PVs specified for the traces, the valid ones will still plot. See notes for X-Y Graph.

Older optimized version of Bar	caThermo	The ‘Older optimized version of Bar’ widget is an older version of the widget that had extremely fast drawing performance. It is used in some LCLS EDM panels to show very quickly updating values, like 10 Hz data from BPMs. The caThermo widget is much slower to draw than this fast bar widget.
Symbol	None	The Symbol class lets you define an array of groups of graphics primitives, and display a different item from that array based on the value of a PV. It is used in many LCLS panels, to graphically represent things like a PPS door. There is no straightforward caQtDM counterpart, although the functionality can be duplicated with many stacked caFrames containing caGraphics items, and visibility calculations on each caFrame.
Animated Symbol	None	An animated version of the Symbol class, where there are additional PVs that control the position and rotation of the symbol. There is no caQtDM counterpart for this widget, but it is unused in any LCLS EDM Panels.
Text Control	caTextEntry	The Text Control widget lets you click on the text to edit it. A pop-up dialog box is displayed that lets you change the value of the PV. caQtDM doesn’t have a widget that duplicates this exact behavior, but it does have caTextEntry, which is a standard text entry field.
Text Entry	caTextEntry	
Slider	caSlider	The EDM Slider widget has a few features that the caSlider doesn’t duplicate: There is an option for a built-in save and restore button in the EDM version, and a built-in readback label. These features could be recreated in caQtDM by using other widgets in concert with the slider. Visually, the EDM slider looks very different than caSlider.
Motif Slider	caSlider	
Button	caMessageButton or caToggleButton	The EDM Button widget has two modes: Push and Toggle. To duplicate a Push mode button use caMessageButton. To duplicate a Toggle mode button use caToggleButton. In EDM, if you use the button with an enum PV, it assumes the enum has a 0 value and a 1 value. caQtDM doesn’t make that assumption. In both modes, the EDM Button has a feature where the button’s label text is the Enum string for the PV. This feature does not exist in caQtDM. More investigation needs to be done to see if these differences will be a big stumbling block in converting EDM panels.
Menu Button	caMenu	
Radio Box	caChoice	
Message Button	caMessageButton or caToggleButton	EDM’s Message Button has a toggle mode and a push mode. To get push mode behavior, use caMessageButton. To get toggle mode behavior, use caToggleButton. Message Button has a few features that don’t exist in caQtDM, like ‘Close Display on Press’.
Up/Down Button	None	EDM’s Up/Down button lets you increment or decrement the value of a PV with left and right mouse buttons. It is only used in four LCLS panels. This button is confusing to use, and probably shouldn’t be duplicated.
Ramp Button	None	EDM’s Ramp button continuously changes the value of a PV from a minimum value to a maximum. There is no caQtDM counterpart. This button is only used in one LCLS EDM Panel. The ramp button functionality could be duplicated with a shell script plus a caShell-Command button.
Freeze Button	None	The Freeze Button freezes all activity on an EDM panel. There is no caQtDM counterpart. This button is rarely used in LCLS panels, and is usually just a convenience to make it easier to take a screenshot of the panel.

Exit Button	QPushButton	The EDM Exit Button closes the current panel. It also has an option to close the parent panel, if it is used in an embedded window, and can also exit the whole EDM program. It is used on a huge number of LCLS panels, but the functionality is really more of a minor convenience than an essential feature. The user can just close the window with the standard window ‘close’ button (the ‘X’ button at the top of the window), or use a QPushButton, and connect the clicked() signal to the parent widget’s close() slot.
Menu Mux	None	Menu Mux is used in many panels, and there is no direct replacement for it. Menu Mux lets you create a drop-down menu with multiple choices. Each choice sets values for multiple macro variables. caQtDM’s manual mentions making “soft PVs”, which you could use along with the caMenu widget to get most of the same functionality, but not all of it.
Related Display	caRelatedDisplay	
Shell Command	caShellCommand	
Choice Button	caChoice	
Multi-line Text Entry	caTextEntry	The Multi-line Text Entry widget lets you change a multi-line text string PV. This widget is used in one LCLS EDM panel (a test panel for LI28 LLRF), and on that panel it could be replaced with a single line text entry. So, for our purposes, caTextEntry is a fine replacement option.
Triumpf Slider	QEAnalogSlider	Unclear what the difference is between Triumpf slider and Motif slider. caSlider is a good replacement for the Motif Slider (see above notes for Motif slider).
Multi-Slider	None	Works like the EDM Slider class, but has controls for two PVs on the same slider. Only used on a couple of EDM panels for oscilloscope control. Could be replaced with two separate caSliders.

TABLE III: Analysis of the caQtDM widget set, as compared to EDM.

### Appendix C: CSS Widget Analysis

EDM Widget	CSS Counterpart	Notes
Lines	Polyline	
Rectangle	Rectangle	
Circle	Ellipse	
Arc	Arc	
Static Text	Label	No ‘Alarm Sensitive’ option. Might be possible to do it with a script.
Text w. Reg. Exp.	caLabel	Text w. Reg. Exp. lets you show or hide a text label based on a regular expression evaluation of the label’s contents. This widget is used in a large number of LCLS EDM panels, but the regular expression feature is not used by any LCLS panels. In all SLAC use-cases, this could be replaced by CSS Label.
Embedded Window	Linking Container	
PNG Image	Image	
GIF Image	Image	
Dynamic Symbol	None	The Dynamic Symbol class lets you build simple animated graphics objects out of graphics primitives, like lines, rectangles, etc. This widget has no CSS counterpart, but it is not used in any LCLS EDM Panels, so this isn’t a big issue.
Meter	Meter	
Bar	Progress Bar	
Message Box	None	The EDM Message Box widget has no CSS counterpart. This widget is only used in one LCLS EDM Panel, which was used for vacuum comissioning purposes. Not essential for SLAC’s purposes.

X-Y Graph	XY Graph	
Text Monitor	Text Update	Cannot choose a separate PV for alarm coloring.
PV Inspector	No widget, built into CSS	There is no CSS replacement for the PV Inspector widget, but a PV Inspector-like feature is built into CSS.
Table	None	The EDM Table widget uses a PV as a filename, and tries to open a text file, and display the contents of that file as a table. There is no CSS replacement. This widget is not used in very many panels, but the panels that have it are used quite frequently (BCS/PPS CUD is a big one).
Coef Table	None	The Coef Table shows the individual values of a waveform PV as rows in a table. There is no CSS counterpart.
Indicator	Progress Bar	Set 'Indicator Mode' to 'yes'.
Text Update	Text Update	Cannot choose a separate PV for alarm coloring.
Reg Text Update	None	The Reg Text Update widget lets you define a regular expression to show or hide the text. While this EDM widget is used frequently at SLAC, the regex feature of it is not used in any LCLS panel, so a conversion to Text Update would work fine.
Byte	Byte Monitor	
Hoff Video	Intensity Graph	
Multi-line Text Update	None	Multi-line text update is a text update that can display multiple lines of text, in the case that it is displaying a long string PV with line breaks. There is no direct CSS counterpart, but the multi-line feature of the EDM widget is not used in any LCLS EDM panels, so it can be safely replaced with Text Update.
SLAC XY Graph	XY Graph	
Older optimized version of Bar	Progress Bar	The 'Older optimized version of Bar' widget is an older version of the widget that had extremely fast drawing performance. It is used in some LCLS EDM panels to show very quickly updating values, like 10 Hz data from BPMs. The Progress Bar widget is not as fast as the EDM widget, but it is close.
Symbol	Multistate Symbol Monitor	CSS' Symbol classes use images (PNG, GIF, etc) for each state, rather than graphics primitives, which would make automatic conversion a little hard.
Animated Symbol	None	An animated version of the Symbol class, where there are additional PVs that control the position and rotation of the symbol. There is no CSS counterpart for this widget, but it is unused in any LCLS EDM Panels.
Text Control	Text Input	The Text Control widget lets you click on the text to edit it. A pop-up dialog box is displayed that lets you change the value of the PV. CSS doesn't have a widget that duplicates this exact behavior, but it does have Text Input, which is a standard text entry field.
Text Entry	Text Input	
Slider	Scaled Slider	The step size cannot be changed at runtime. Does not have save/restore buttons.
Motif Slider	Scaled Slider	Step size cannot be changed at runtime.
Button	Action Button	
Menu Button	Menu Button	
Radio Box	Radio Box	
Message Button	Action Button	Need to use scripts or button actions to duplicate all the EDM Message Button's features.
Up/Down Button	Spinner	EDM's Up/Down button lets you increment or decrement the value of a PV with left and right mouse buttons. Spinner kind of serves the same purpose - a widget that lets you step a PV up and down by fixed amounts. Spinner is much better than Up/Down button though, as it has a built-in indicator, and doesn't require the user to know in advance that you need to right click to increment and left click to decrement.

Ramp Button	None	EDM's Ramp button continuously changes the value of a PV from a minimum value to a maximum. There is no CSS counterpart. This button is only used in one LCLS EDM Panel. The ramp button functionality might be possible to duplicate with a script on an Action Button widget.
Freeze Button	None	The Freeze Button freezes all activity on an EDM panel. There is no CSS counterpart. This button is rarely used in LCLS panels, and is usually just a convenience to make it easier to take a screenshot of the panel.
Exit Button	Action Button	Need to use a script on an Action Button to close the display.
Menu Mux	None	Menu Mux is used in many panels, and there is no direct replacement for it. Menu Mux lets you create a drop-down menu with multiple choices. Each choice sets values for multiple macro variables. Potentially, widget scripts could be used to replace this functionality.
Related Display	Action Button	Use the 'Add Open OPI' button action.
Shell Command	Action Button	Use the 'Add Execute Command' button action.
Choice Button	Choice Button	
Multi-line Text Entry	Text Input	The multi-line feature of this EDM widget is unused at SLAC. CSS does duplicate the functionality, though.
Triumph Slider	Scaled Slider	See the comments above for the EDM Motif Slider.
Multi-Slider	None	Works like the EDM Slider class, but has controls for two PVs on the same slider. Only used on a couple of EDM panels for oscilloscope control. Could be replaced with two separate Scaled Slider widgets.

TABLE IV: Analysis of the CSS widget set, as compared to EDM.