

Conversion of Hall D CSS-BOY Screens to WEDM

Tyler Lemon, Mary Ann Antonioli, Peter Bonneau, Aaron Brown, Pablo Campero, Brian Eng, George Jacobs, Mindy Leffel, Marc McMullen, and Amrit Yegneswaran

Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606

August 20, 2019

To facilitate remote monitoring, the Detector Support Group (DSG) converted CSS-BOY screens that are critical for Hall D operations to Web Extensible Display Manager (WEDM). This note discusses the CSS-to-WEDM conversion process.

Hall D uses CSS-BOY screens to interface with the EPICS controls and monitoring system. To access CSS screens, a remote connection through hallgw, the gateway through accelerator’s network firewall, has to be established using two-factor authentication followed by an additional connection to a Hall D controls server. While this is a reliable access method, it is cumbersome as it requires users to access the monitoring screens through a terminal interface with remote desktop capabilities. To remedy this situation, DSG converted the CSS-BOY screens that are critical for Hall D operations to Web Extensible Display Manager (WEDM), which allows users to access the screens from any phone, tablet, or web browser.

WEDM displays screens as HTML, enabling them to be viewable by a web browser. WEDM is maintained by the Jefferson Lab Computer Center and requires a CUE login to access the screens. To further increase security, WEDM is read-only with no control capabilities.

In all, nineteen screens were converted to WEDM:

- chiller (BCAL, ComCal, and FDC)
- solenoid (power supply, strain gauges, interlocks, voltage taps, vacuum pumps, cryogenics, and temperatures for coils 1–4)
- environment (ComCal, DIRC, and Hall D ambient)
- gas system (CDC/FDC)
- tagger NMR control
- beamline vacuum.

To display a CSS-BOY screen in WEDM, the CSS screen must first be converted to Extensible Display Manager (EDM). Both display types use text-based file formats to assign properties to items on the screen. CSS’s *.opi* file format is a form of XML, while EDM’s *.edl* file format is plain text. To be able to extract screen properties from CSS and create equivalent EDM screens, a Python script was developed.

The Python script uses an XML parsing package to extract widget properties from the *.opi* file and then uses templates for EDM widgets to place equivalent items on the EDM screen. The current version of *opi2edl* supports the conversion of static text (labels), lines, circles, rectangles, GIFs, PNGs, bar monitors, text monitors, and LEDs.

The conversion script successfully converted the basic widget properties for all screens. Because WEDM is read only without controls capability, the conversion script does not place controls widgets; this generated a significant amount of empty space on the converted WEDM screens. Additionally,

CSS and EDM have different default text sizes, which also leads to excess empty space on converted screens; this issue was fixed by manually editing the resulting EDM screens to remove empty space previously occupied by controls and re-sizing/arranging text. Figure 1 shows a CSS screen, Fig. 2 the EDM screen result, and Fig. 3 the manipulated screen.

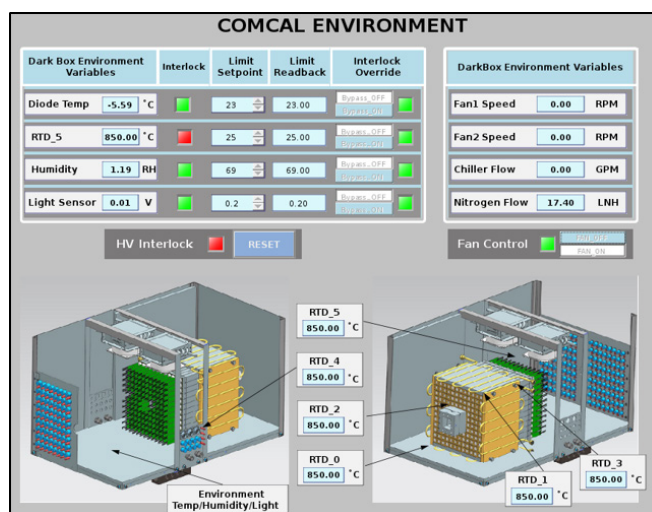


FIG. 1. ComCal Environment CSS-BOY screen.

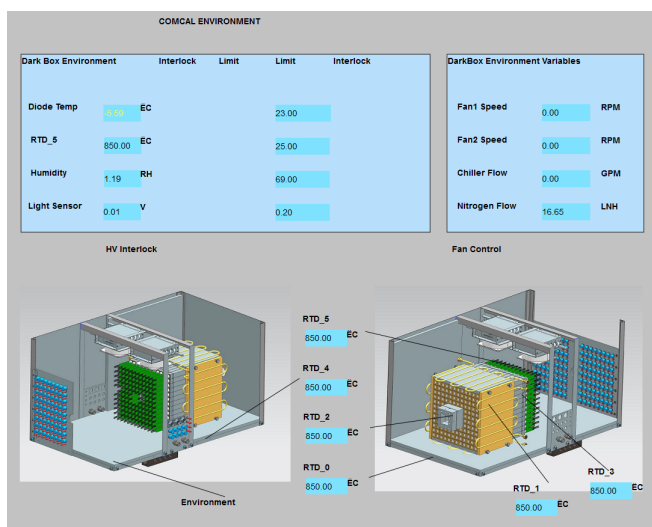


FIG. 2. ComCal Environment EDM screen immediately after conversion from CSS to EDM using *opi2edl* script. Note the empty space left on screen by missing controls and text size differences.

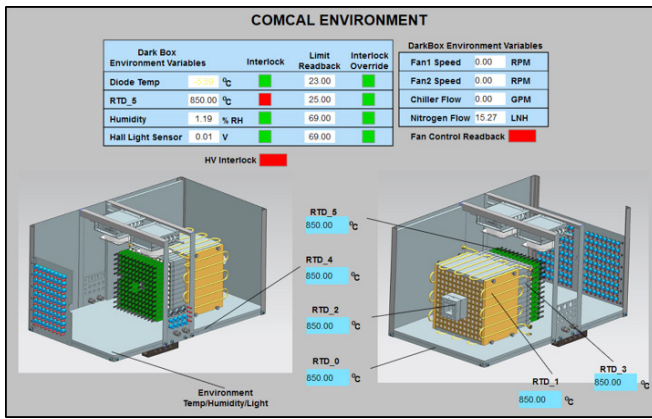


FIG. 3. ComCal Environment EDM screen deployed to WEDM server. Note that items on screen have been rearranged to remove empty space and text has been appropriately resized.

While CSS can execute embedded JavaScript programs to create local process variables (PVs), EDM, and by extension WEDM, cannot, which led to screens not displaying items whose PVs are generated by scripts in CSS. However, if the local PV generated by CSS is the result of a simple operation (addition, subtraction, or multiplication), WEDM can use a CALC PV type to generate a local PV to display the value, as was the case for the summation of voltages for the solenoid coils. For more complex scripts, such as the one used to parse an array for the solenoid power supply status bits, new PVs had to be added to the Hall D IOC. Only then were the PVs able to be displayed in WEDM.

CSS has capabilities so that widgets can graphically appear as thermometers or gauges. Unfortunately, WEDM does not have equivalent widgets, so on screens that used graphical CSS widgets, text monitors were placed on WEDM screens instead (Fig. 4).

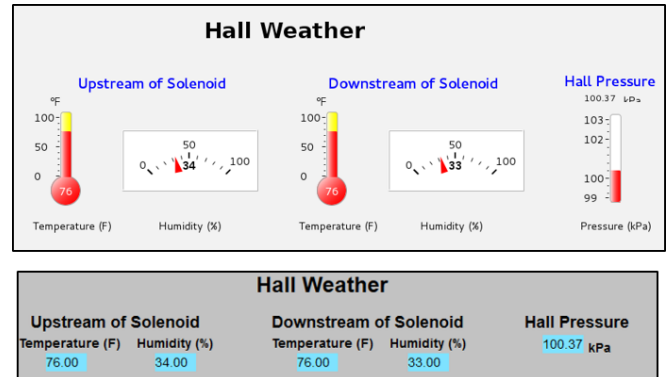


FIG. 4. Top: CSS screen for hall environment using graphical thermometer and gauge widgets. Bottom: WEDM screen showing graphical widgets replaced by text monitors.

In summary, to facilitate easier off-site monitoring of critical Hall D systems, CSS screens were converted to EDM using a DSG-developed script and then loaded into Accelerator's WEDM server. Nineteen screens have been successfully converted to WEDM and links to them added to the epicsweb main menu (<https://epicsweb.jlab.org/wmenu/#HallDDetectorComponents-page>).