

ADAPTATIONS TO CS-STUDIO FOR USE AT DIAMOND LIGHT SOURCE

W. Rogers, N. Battam¹, T. Cobb, M. Furseman, G. Knap
Diamond Light Source Ltd., Oxfordshire, UK
¹Tessella Ltd., Abingdon, UK

Abstract

Control System Studio [1] (CS-Studio) is one of the most widely-used display managers for EPICS. It is based on the Eclipse Rich Client Platform (Eclipse RCP), allowing for coherent integration of interfaces for different systems with common graphical elements and preferences. However, this user interface presents a different way of working to those from the previous generation of EPICS tools such as Extensible Display Manager [2] (EDM) and Striptool. At Diamond Light Source, EDM has been used since commissioning in two different ways: for machine operations and for beamline controls. Both uses of EDM will eventually be replaced with CS-Studio and significant effort has been put into this transition. Two kinds of change proved necessary: adaptations to CS-Studio itself, and changes to the typical user workflows. This paper presents both types of changes that were needed to make CS-Studio a productive tool at Diamond.

INTRODUCTION

Since commissioning in 2007, Diamond Light Source (DLS) has used the display manager EDM for all graphical interfaces to the EPICS control system, for controlling both the accelerators and the photon beamlines. EDM has performed well, but now relies on libraries that are becoming obsolete, and is supported by only one developer. CS-Studio is one of the most widely-used replacements for EDM. It is written in Java and based on Eclipse RCP, and so it runs on Windows, MacOS and Linux, but this also means that it has a very different user interface to EDM. Making the transition between the two applications presented a number of challenges.

MODIFICATIONS TO THE CS-STUDIO PRODUCT

Since CS-Studio is an open-source collaboration, users are free to modify the code for their own purposes. DLS has decided that the benefits of working with the CS-Studio collaboration to maintain a single code-base outweigh the freedom to fork the code and make any convenient changes. Working with the collaboration, a number of changes to the CS-Studio application have been made that helped ease the transition from EDM to CS-Studio.

Automatic Conversion

Many thousands of EDM displays are in use at DLS. To reproduce this behaviour in CS-Studio, it is essential that those displays can be automatically converted into equivalent screens, since it would not be feasible to recreate all of these screens from scratch. EDM defines its displays in a custom

file format that describes the layout of a number of types of widgets and the properties of each widget. CS-Studio's display tool BOY uses an XML-based format that describes layouts in a similar way, with a range of widgets that are similar but not identical. At the time that DLS began to move to CS-Studio, a tool existed to make a widget-for-widget conversion between the two formats. We have put a significant amount of effort into improving the quality of conversion, including creating new widgets to allow more accurate reproduction of the same behaviour. The converted screens are in most cases difficult to distinguish from the originals rendered by EDM, and other members of the community will benefit from the improved conversion capability.

Standalone Windows

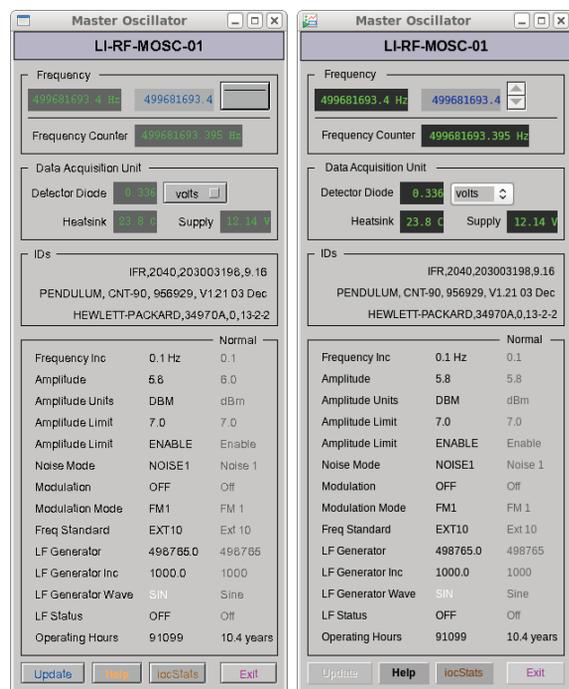


Figure 1: An EDM screen (left) and the same screen converted for CS-Studio (right).

EDM operates using many individual windows, and the DLS operators have defined a way of working that lays out many such windows across multiple Linux virtual desktops, in a classic control-system-style interface. CS-Studio, because of its use of the Eclipse toolkit, shows its content inside one or more large windows known as workbench windows, with a tabbed interface allowing rearrangement of a number of views inside those windows. Because the control

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

system interfaces are laid out with fixed sizes and locations, arranging those screens inside a larger window is difficult.

A new feature was added to CS-Studio to allow screens to launch in standalone windows in much the same way as EDM. This allowed operators to use the same way of working as before, and meant that exactly the same layout of screens could be used. Since this behaviour is different to the standard behaviour of Eclipse, care was taken to make the integration as intuitive as possible, including adding a new view in the workbench to allow locating the standalone windows. A screen rendered in EDM and the corresponding converted screen, in a standalone window, are compared in Fig. 1.

When a user is using only standalone windows, or is not using CS-Studio directly at any time, it can be useful to minimise the main window and leave CS-Studio running in the background. This helps to mitigate the relatively long startup time for the application. A new feature allowing this behaviour has also been added to the CS-Studio product (see Fig. 2). When an action is taken on a standalone window that requires use of the main window, the main window will present itself appropriately.



Figure 2: CS-Studio minimised to the system tray on Linux.

Filesystem Layout

The DLS Controls Group has a well-established system for installing production software. Each version of a software product is installed on a read-only filesystem. In order to keep EPICS displays up to date with the IOCs and support modules with which they are used, these displays are contained within the same code repositories and released at the same time. Furthermore, the existing DLS release process allows different IOCs to depend on different versions of the same support module.

Eclipse and CS-Studio have their own way of understanding filesystems: the files that they use are expected to be contained inside certain directories known as ‘workspaces’, alongside user preferences and other metadata. This layout is incompatible with the relevant files being distributed across the read-only filesystem.

To resolve this problem, we extended an existing Eclipse capability to link a directory inside a workspace to an external directory in a different location. When this has been done, the external directory is available for use by CS-Studio. If a display in one module needs to launch a display contained in another (for example, screens in many IOCs link to the screens in the *motor* support module), these links are set up so that each IOC screen will open the correct version of the motor screen.

Launching Screens from Outside the Application

Most applications used by the DLS Controls Group are launched using a tool called the Launcher, shown in Fig. 3. To support this way of working, it needed to be possible to update filesystem links (see the previous section *Filesystem layout*) and launch displays using an external command. Eclipse supports this behaviour to some extent, but a number of modifications were required to ensure that this process worked reliably.

When a user launches a beamline environment, they need to open a set of screens inside a main CS-Studio window. To get predictable behaviour a new CS-Studio window is opened with a custom beamline layout. A new file format was defined that specifies how to open this new window. Each beamline may have its own defined layout and can be opened from an entry in the Launcher. One such layout is shown in Fig. 4.

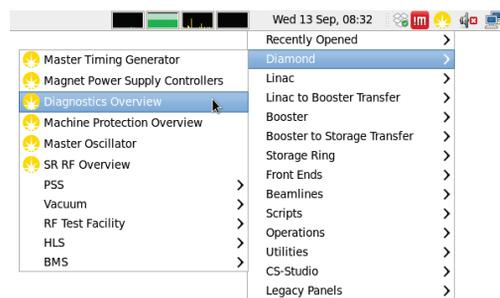


Figure 3: The DLS Launcher.

CHANGES TO USER WORKFLOWS

During this transition process, much effort was put into ensuring that users would not have to change the way they worked. This made the transition significantly smoother. In some cases, however, the change in the tools being used led to a more efficient way of working and an improved user experience.

Eclipse-Style Usage

While the Operations Group at DLS has a well-established and efficient way of working with EDM that we did not wish to change, the Beamline Controls Group identified ways that beamline users found working with EDM difficult. Use of the beamline screens typically led to a large number of EDM screens open and no easy way to find the one that is needed.

In this case, using CS-Studio in a similar way to Eclipse allows more control over the layout of these panels. A main window is divided into three parts: a main pane in the top left, a ‘synoptic’ view along the bottom and a details panel on the right. Overview panels open in the main pane, and further details open in a tabbed view to the right. This layout means that when you open a screen it will appear in the expected location; the tabs mean that many previously-opened screens do not distract the user.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2017). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

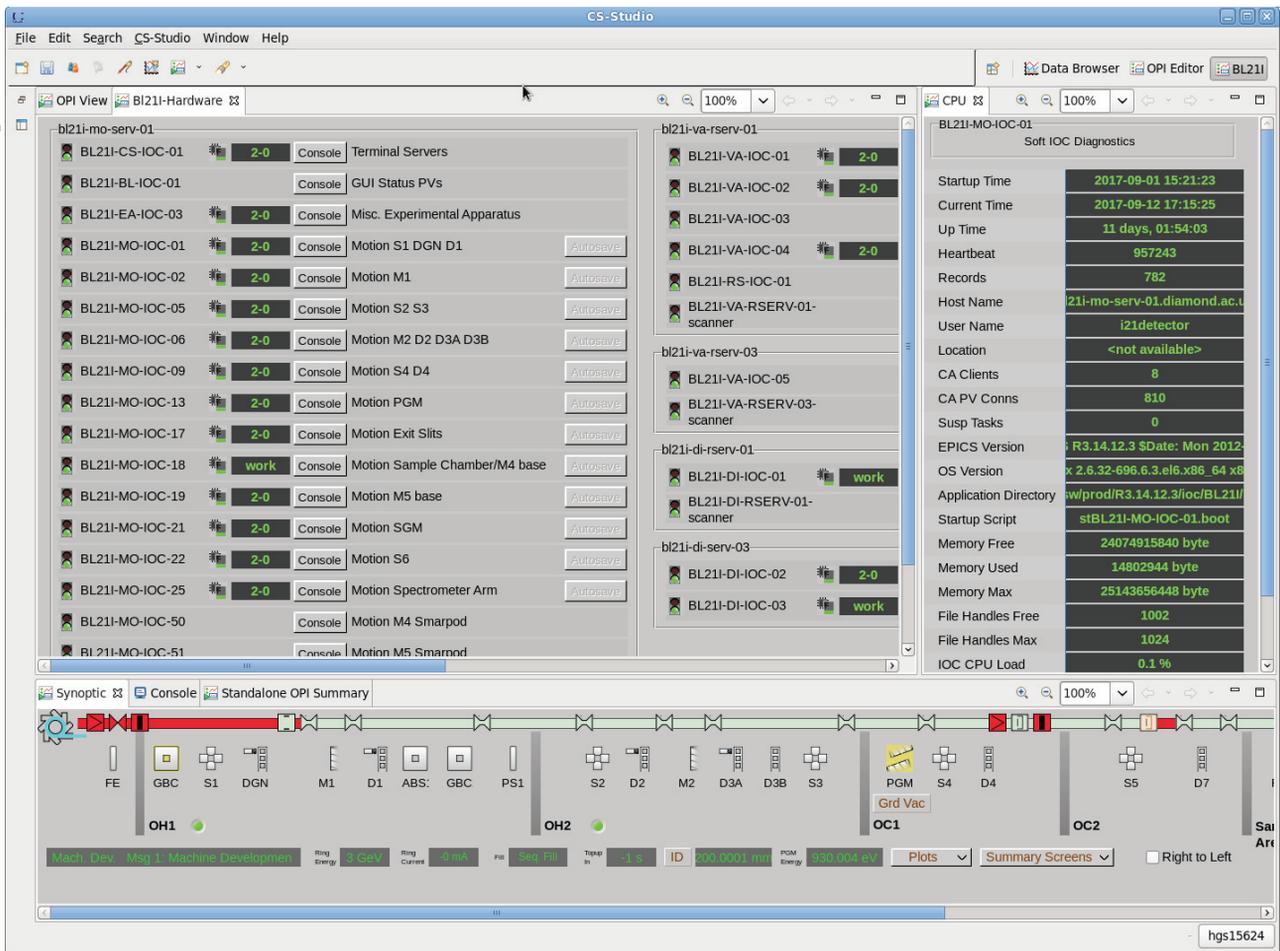


Figure 4: The BL21I beamline screens, using a layout similar to a standard Eclipse RCP application.



Figure 5: Databrowser plot showing EPICS archive data for a corrector magnet and fast archive data for nearby BPMs. The effect of the top-up process can be seen at ten-minute intervals.

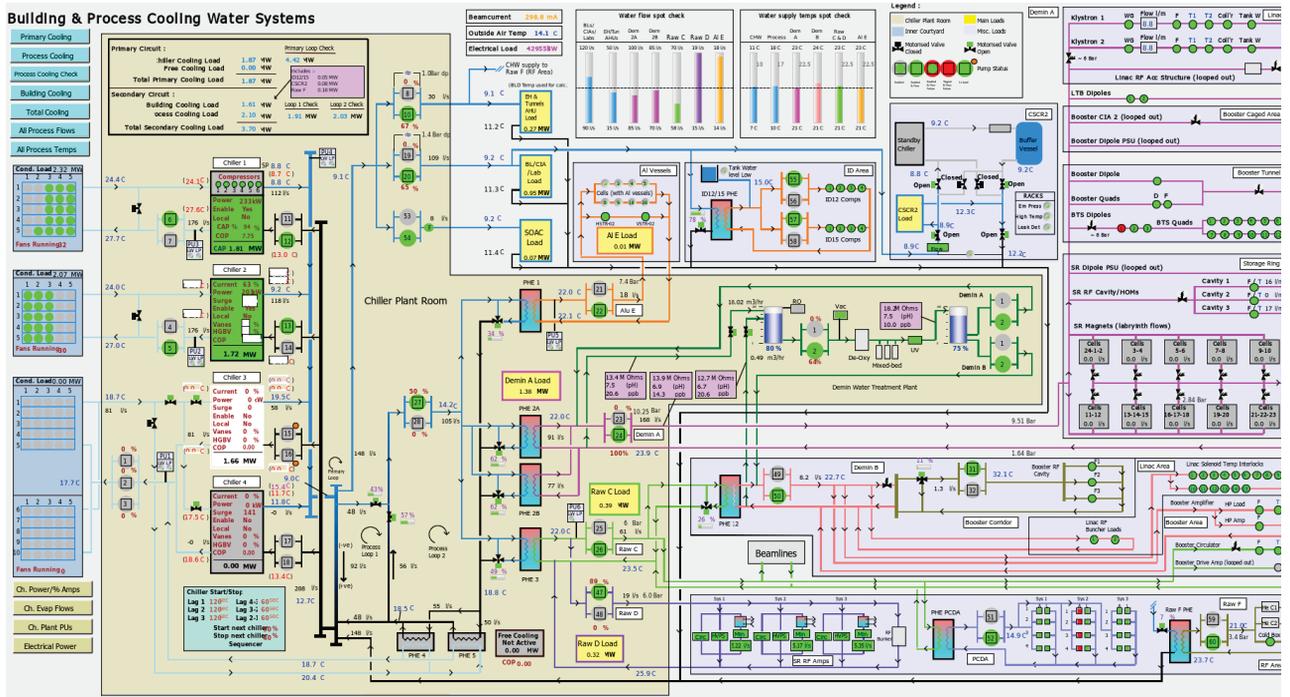


Figure 6: Overview of the BMS water cooling system at DLS.

Databrowser

The Databrowser tool in CS-Studio allows viewing data retrieved from archives. At DLS there are two different archivers in use: a Fast Acquisition Archiver [3] that stores 10kHz data from electron and photon BPMs, and the EPICS Channel Archiver that stores data from EPICS at a rate of up to 10Hz. These archivers have their own tools that allow displaying the data.

CS-Studio includes support for the Channel Archiver in the Databrowser tool, which is a more capable and intuitive tool than the Channel Archive Viewer. Since the Databrowser was designed to be extensible, it was possible to add support for the fast archiver to the same application. This allows displaying data from both archivers on the same plot for the first time. An example is shown in Fig. 5.

The community support for CS-Studio and extensibility of Databrowser also means that the same application can be used for the newer archiver that is planned for deployment at DLS, the EPICS Archiver Appliance [4]. This will reduce the disruption when making that transition.

New Users

Since CS-Studio runs on Windows and Mac as well as Linux, it has allowed new users at DLS to start using EPICS. The Building Management Services (BMS) previously used a proprietary control system. Using this system it was difficult to visualise the whole BMS or large subsystems on one screen. Parts of the BMS are now mapped to EPICS process variables, and now newly-developed visualisations are in use by the Installation and Facilities Management Group and

in the Control Room by the Operations Group. The BMS overview screen is shown in Fig. 6.

CONCLUSIONS

The DLS Controls Group is in the process of moving all EPICS displays to use CS-Studio. The large number of screens that are used and the differences between the two applications make this a challenging process. By a combination of improving CS-Studio itself, giving it new features, and user training, we have made CS-Studio a suitable application for all uses at Diamond, and we have improved the user experience in several ways.

By the end of 2017 all displays used to control the accelerators will be using CS-Studio. Beamlines will move from EDM to CS-Studio one at a time over the coming year.

REFERENCES

- [1] J. Hatje, "Control System Studio", in *Proc. ICALEPCS'07*, Knoxville, USA, Oct. 2007.
- [2] Extensible Display Manager, <http://ics-web.sns.ornl.gov/edm>
- [3] M. G. Abbott, "A New Fast Data Logger and Viewer at Diamond: the FA Archiver", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011.
- [4] M. Shankar, L. Li, M. Davidsaver, M. Konrad, "The EPICS Archiver Appliance", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015.