Table of Contents

1 Introduction: what is Drivestat?

DriveStat is a newly developed application written in JavaFX which makes use of JAPC, a unified Java API for Parameter Control relating to equipment access. JAPC was developed at CERN and then adapted at GSI. DriveStat allows the user to perform an online monitoring of the status of drives belonging to a selected beamline from the ion source up to the experiment. Within the document, drives will be often called as “devices”. They can be stepper motors (DS) and pneumatic drives (PLA).

2 How to start the application

In order to start DriveStat, open a new terminal and perform the following instructions:

- Type ”`scheidenb@asl740.acc.gsi.de`” followed by password to connect to an “asl” machine.
- Navigate to “`/common/usr/cscoap/bin`”.
- Start the launcher typing “`./launcher pro`”.
- Select the tab “FRS”.
- Select “FRS DriveStat” from the menu to start the application.

3 The working principle

Figure 1 depicts a simple schematic describing the working principle of DriveStat. Data treated by the application come from the following sources:

- FESA (Front End Software Architecture) database, related to hardware devices.

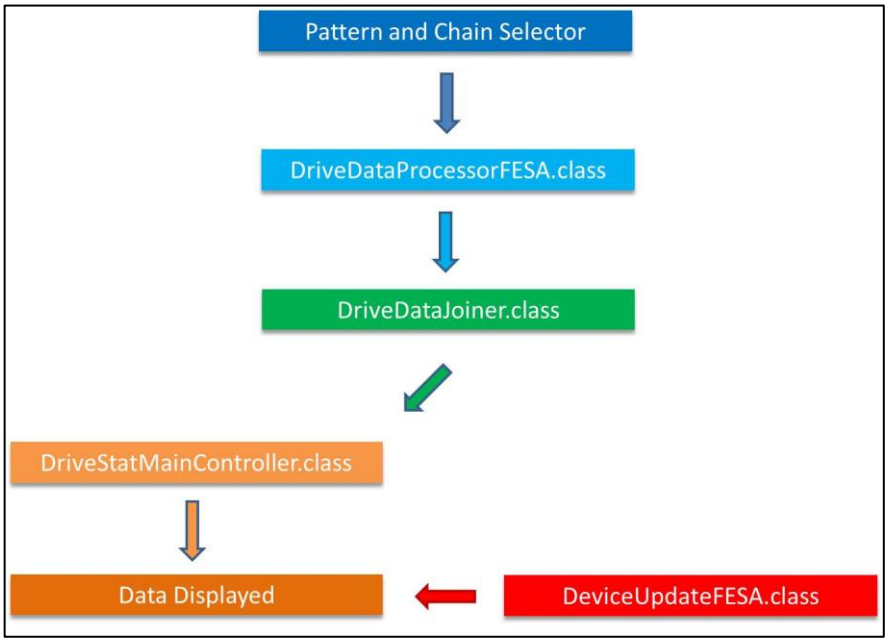

Figure 1: schematic of the DriveStat working principle.

DriveStat executes a sequence of tasks which can be summarized as follows:

- *DriveDataProcessorFESA.class* retrieves, from the FESA database, a list of devices belonging to a certain beamline chosen via the Pattern and Chain selectors (see Section 4.1). For each device, a set of actual parameters (see Table 1) is loaded.
- *DriveDataJoiner.class* creates a structure of data. The latter are injected to the *DriveStatMainController.class* and then displayed in the table as shown in Figure 1.
- *DeviceUpdateFESA.class* starts an online monitoring of the FESA parameters values. In this case, the update is performed whenever a change in any of the parameters value is detected.

4 The main user interface and its components

Figure 2 shows the DriveStat main user interface. It is divided into four regions each of which contains the following items:

- Top: pattern and chain selectors and update Drivese check box;
- Center-Left: the table for online monitoring of the devices status;
- Right: action buttons;
- Bottom: status bar for messaging including a data saving LED indicator and a loading progress bar which activates whenever new data are requested.

Table 1: the list of parameters retrieved from the FESA database.

Parameter	**Description**
POSIABSI/data_pos	It describes the absolute position value of drives type DS.
CONSTANT/minPosition	It represents the minimum position assumed by a DS drive.
CONSTANT/maxPosition	It represents the maximum position assumed by a DS drive.
POSITI/position	It provides the position of a PLA drive.

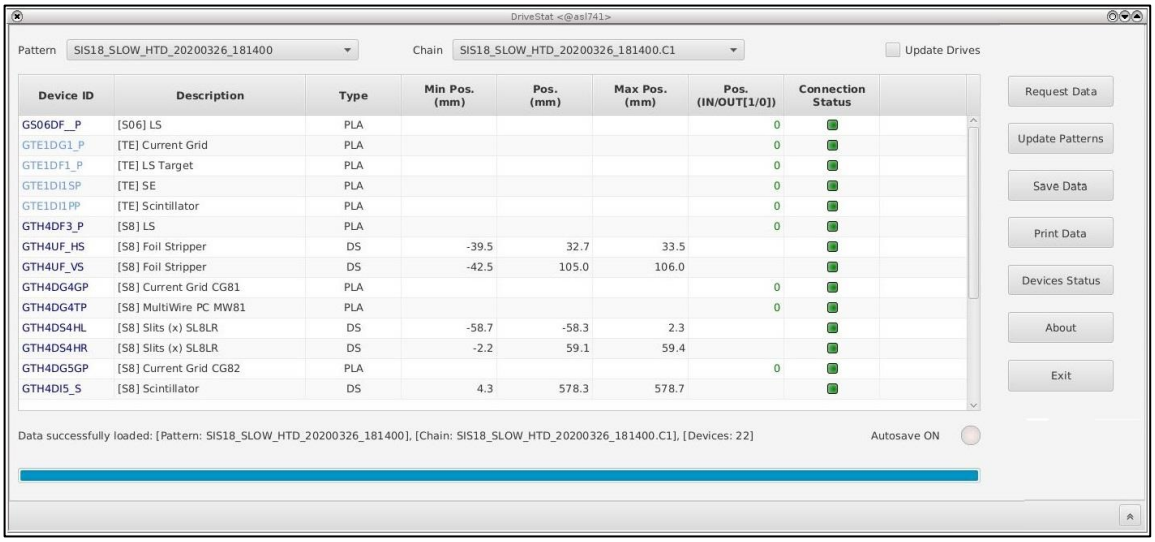

Figure 2: the main user interface.

4.1 Pattern and Chain selectors

Pattern and Chain selectors contain a list of available patterns (both active and inactive) and chains. A pattern represents a group of beam production chains that are executed periodically in a certain order (so far each pattern has only one beam production chain associated). A beam production chain defines a sequence and parameters of beamlines.

A beamline is selected by choosing a combination of pattern and chain. If devices related to the selected pattern and chin are available, their associated parameters (see Table 1) will be loaded and the relative values displayed in the table as shown in Figure 2. If the loading process terminates with no errors, a message in the status bar will indicate the total number of loaded devices.

4.2 Update Drives check box

By ticking the update Drives check box, it will be possible to send and set new position settings of the drives. This is executed by the method *runSequence* built in the SetAllFRSDrives.class. For safety reasons, the user will be asked to provide a password before starting the updating process. Once the access has been granted, updates are performed every 60 seconds. Grant is valid only for the current set of loaded drives. For data relative to a new combination of pattern and chain, password will be required again.

4.3 Table

The main table consists of eight columns each of which containing the following information:

- **Device ID**: it represents the unique name which identifies a drive.
- **Description**: it describes the functionality of each device. At the moment, such information is contained within DriveStat and therefore retrieved internally. In the next release, functionality of each drive will be directly imported from an external database which is entirely managed by the user through the new application DevConf (for more information, refer to the manual).
- **Type**: it assumes two values: *DS* for stepper motors and *PLA* for pneumatic drives.
- **Min Pos.**: it shows, in mm, the minimum achievable position of a DS device.
- **Pos.**: it reports the actual position of a DS device. Measurement is shown in mm.

- **Max Pos.**: it shows, in mm, the maximum achievable position of a DS device.
- **Pos.**: it represents the actual position of a PLA device. It assumes only two values: 0 and 1. A value of 0 indicates that the device is out the beamline while 1 means that the device is inserted in the beamline.
- **Connection Status**: it shows whether a drive is properly connected. A green box indicates that connection is established and it is possible to retrieve updates regarding the actual position of the device (both DS and PLA type). A red box points out an interruption of the communication with the drive. The communication is refreshed every 10 seconds. In addition, every 300 seconds, data shown in the table are reloaded and a new communication is generated from scratch for each drive.

The table shows data sorted by device location from the synchrotron SIS down to the experimental area of interest.

4.4 Action Buttons

4.4.1 Request Data

It starts the loading process of data related to devices belonging to the selected pattern and chain. A progress bar (see Section 4.5) located at the bottom of the main user interface, indicates the status of the loading process. Only devices included in the DevConf database (see the related application manual) will be displayed in the table. If the database is not available, a default list of devices will be loaded.

4.4.2 Update Patterns

It updates the list of the Pattern and Chain selectors. To start loading data, the Request Data button has to be pressed.

4.4.3 Save Data

The Save Data dialog is shown in Figure 3. Two are the modalities by which data can be saved on disk:

- **Manual**: the user is allowed to edit the default path and the file name. The former is initially set to `/home/rifr/scheidenb/lnx/drivestat`. Use of special characters is prevented for both fields. The button **Save** will be enabled only once the two fields are filled in. Such a modality allows saving data just once. Each saved file will follow a pattern name like `<filename>_<date>_<time>` where the system date and time will be automatically added during the data saving.
- **Auto**: the user is allowed to change the default directory and the frequency at which files should be saved. The default value is 60 seconds although it is possible to decrease it to up to one file saved per second. The **Start** button allows starting the auto saving modality and it will be enabled only once the file path and frequency fields are filled in. The **Stop** button allows the interruption of a previously activated auto saving modality. In this case file will follow a pattern name like `DriveStat_file_<date>_<time>` where the system date and time will be automatically added during the data saving.

Figure 3: the Data Save dialog.

In case the chosen file path doesn't exist, a message error will be prompted. By closing it, the Save Data dialog will continue to remain open. The **Close** button in the Save Dialog allows the user to return to the main user interface.

Figure 4 shows the structure of a saved file. The first line reports the selected pattern and chain while the second one the name of the optics adopted. Next it follows the header with the data acquired from the DriveStat main table. Data reported in the "Conn. Status" assume value of 0 and 1. A value of 1 indicates an error in the hardware device connection. In this case, the position value of the relative device might not be correct.

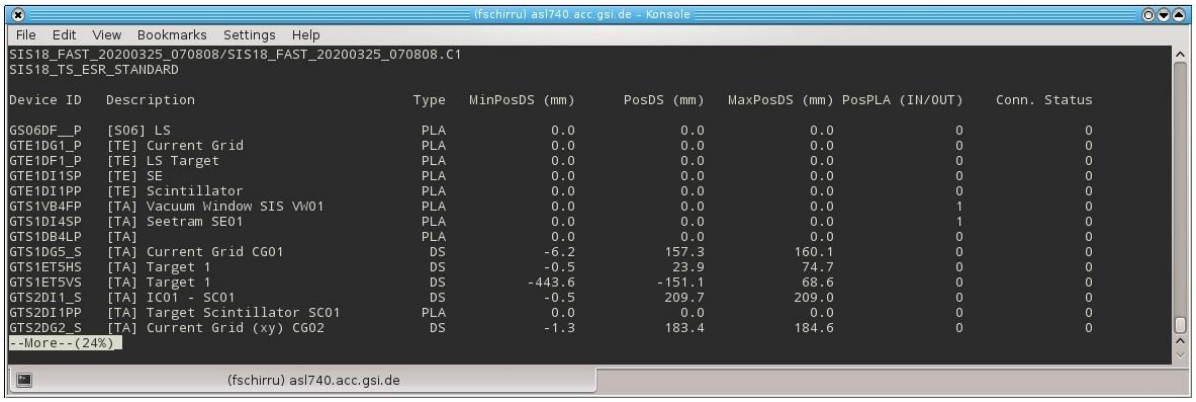

Figure 4: the structure of a saved file.

4.4.4 Print

Figure 5 displays the Print dialog. The function is enabled only if the main table contains a list of devices for a certain selected pattern and chain. Regardless of the number of listed devices, data will be automatically fitted in one A4 page. The Printer selector allows the user to choose one among the printers available within the network. By default, "Printer p293" located in the Messhütte is selected. The **Print** button prints out the data currently displayed in the table while the **Close** button let the user to return to the main user interface.

4.4.5 Device Status

This option allows the user to perform a manual connection to the device hardware in case it is lost. Since the actual version of DriveStat automatically closes and reopens all the connections every 60 seconds, such functionality will be removed in the future versions.

4.4.6 About

As reported in Figure 6, the About dialog provides general information about DriveStat (Version, Authors, Contacts).

4.4.7 Exit

DriveStat and all its related running processes will be closed.

Figure 5: the Print dialog.

Figure 6: the About dialog.

4.5 Status and Progress bars

The status bar displays messages to the user whenever a certain event occurs. It is provided of a red LED located to the right side which switches on to indicate that data are currently being saved on disk with auto saving modality (see Section 4.4.3).

The progress bar will be activated any time a new set of devices related to a selected pattern and chain is loaded. At first, the progress bar shows the progressive loading of the drives with their initial position value. Next, it shows the progressive request of connections to start the position monitoring of each device as performed by the update event listener.