# Internal Software Design Recommendation

When implementing FESA device software a few issues should be considered to help ease testing the software from the beginning on and from different point of views,

Application developers desire permanent access to FESA devices to be able to adjust and test their applications. FESA device installations should be offered permanently in an environment well separated from the accelerator to the applications developers. Since providing dedicated equipment for testing the applications would be too costly, the FESA classes should run without equipment hardware on central servers. This implies that timing information will not be available on these servers. The FESA devices for testing purposes should be permanently installed in a test environment that is clearly separated from the productive accelerator installation. This implies that accelerator timing information will not be available.

The idea is to offer FESA device software as mockup devices. These mockup devices should offer the same set of properties and a partially similar behavior as the productive version which is installed in the accelerator. This does not imply to develop the FESA software twice. Ideally the same FESA class implementation runs together with hardware equipment and within the test environment.

Using the same FESA class implementation in two environments can be achieved by separating the FESA class implementation from the interaction with the equipment: create an adapter layer which implements the interaction with the equipment.

It is generally recommended to

* Never directly access the hardware components in the FESA actions but use an intermediate software layer (“device adapter”) that provides access to the device’s properties (server actions, RT-actions). E.g. provide a procedure/method ‘setGain(float gain)’ which encapsulates writing to the gain-register of the equipment. Vice versa implement functions/methods which provide data read from the equipment.Design FESA device software that uses the timing system in the way that server actions run without the need to use real-time actions. This implies that in the test environment without timing different return values have to be expected.

In addition to the adapter for hardware access, a separate mockup adapter should be foreseen which provides the same interface to the FESA class, but without implementing the interaction with the equipment hardware. Only the interface to the FESA device software needs to be served, that is the mockup adapter only accepts and provides data. A simple use case for the FESA device mockup adapter is setting reference values and retrieving actual values. Actual values may be either the reference values itself or a slight variation.

The intermediate adapter layer provides the properties of the hardware equipment for the FESA device software. When running the FESA device software in production the equipment adapter links the FESA class to the hardware equipment. When running the FESA device software within a test environment the mockup adapter simulates the hardware equipment.

To distinguish the productive from the test environment different devices should be instantiated differently during start-up of the FESA device software in the specific environments,

A full device simulation will not be easily realized and is therefore not expected. However a partial simulation of the main properties will help to test FESA device software at an early stage and to develop application software in parallel.

The concept of mockup adapter should be used to provide FESA devices already in a very early stage of development, before the FESA class implementation is finished. To provide a test environment for applications development should start with the mockup adapter.

The final recommendation for development of FESA device software is:

1. Define the main properties which will be used by the operations applications
	1. Implement a basic FESA class that supports the main properties
	2. In parallel implement a mockup adapter to simulate the equipment
2. Implement the adapter for hardware access
3. Refine the FESA class, enhance the adapters, implement other properties