

5.5 Debugging parameterizing

The parameterization step occurs the first time each container is started. It occurs by running the `.local/bin/parameterize.sh` script on the container. Debugging output from the execution of this script can be found on the container in `/tmp/parameterize*`

Within the `labtainer.log`, you can see the step occur following the log entry that reads: “About to call `parameterize.sh...`”. The parameterizing step is preceded by a copying of the files in the `labtainer-student/lab_bin` directory into the container.

6 Automated assessment of student labs

This section describes how to configure a lab for automated assessment of student work. Note the framework does not require automated assessment, e.g., the “results” of a lab may consist entirely of a written report submitted by the student. Support for automated collection of written reports is described in 4.6 and the use of `COLLECT_DOCS` in the `start.config` file.

The goal of automated assessment is to provide instructors with some confidence that students performed the lab, and to give instructors insight into which parts of a lab students may be having difficulty with. The automated assessment functions are not intended to standardize each student’s approach to a lab, rather the goal is to permit ad-hock exploration by students. Therefore, lab designer should consider ways to identify evidence that steps of a lab were performed rather than trying to identify everything a student may have done in the course of the lab.

Automated assessment is achieved by first generating artifact files while the student works. That is described in the first subsection below. Next, artifacts within those files are identified as described in section 6.2. The values of the resulting artifacts are then compared to expected values, as per section 6.3.

6.1 Artifact files

The files from which artifacts are derived include persistent data, such as system logs and `.bash_history`, as well as timestamped snapshots of transitory data such as stdout of a program. Lab designers can also generate customized artifacts in response to student actions using scripts that automatically execute when selected programs or utilities are executed – or when selected files are accessed. The following paragraphs describe how these artifacts are generated.

The Labtainer framework use of timestamps allows designers to express temporal relationships between artifacts, and thus between events. For example, the designer can determine if two distinct artifacts were part of the same stdout stream. Or if artifacts in the stdout stream from one program were occurring during the invocation of a different program that generated other specific artifacts. The framework also can incorporate timestamps from standard log file formats, e.g., syslog, allowing the designer to determine if some logfile entry occurred during the invocation of a program whose stdout stream contains selected artifacts. As a more concrete example, the use of timestamps allows the designer to determine that a specific web log record occurred during invocation of some program that produced a specific artifact.

6.1.1 Capturing stdin and stdout

Each time the student invokes a selected program or utility, the framework captures copies of standard input and standard output, (stdin and stdout) into timestamped file sets. This is transparent to the student. (Also see the following section for capturing program output other than stdout.) These timestamped file sets, selected system logs, and everything relative to