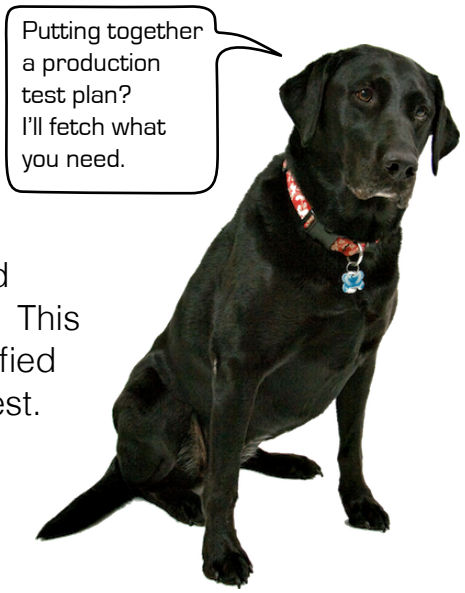
Test Workflow: Putting it all Together



Assembling the elements that comprise a production test draws upon a collection of associated resources: handler configuration files, instrument calibration settings, Fixture definitions, etc. Cassini's use of software objects throughout its system allows for unique organization, management of the test process, and distribution control. Previous documents have introduced the various software and hardware components that make these capabilities possible. This section discusses how these parts come together into a unified workflow to allow a user to coordinate the tasks of production test.

User Roles

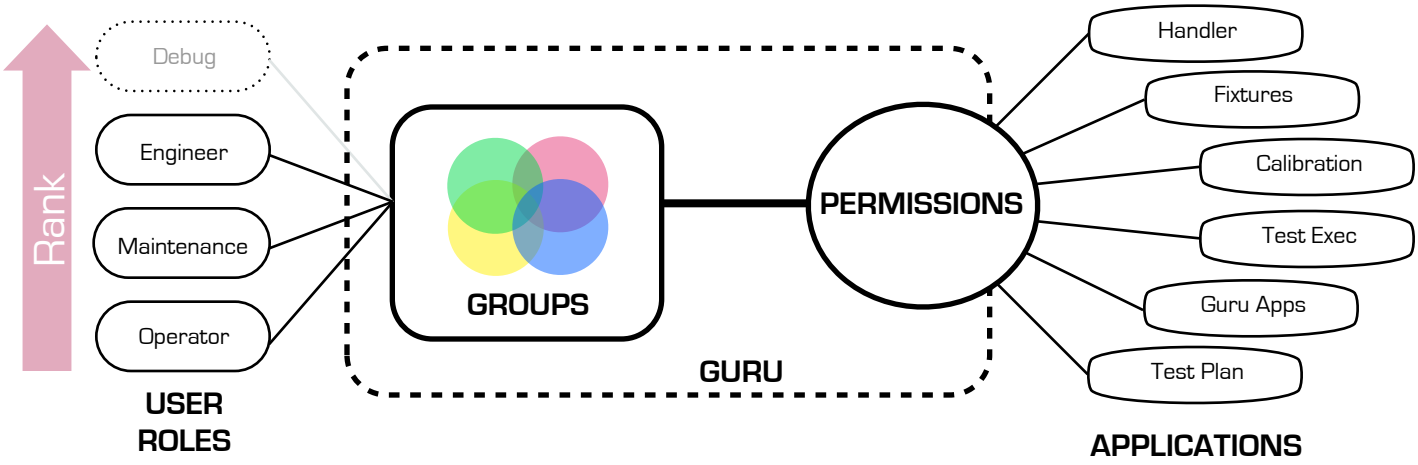
Production test is a large undertaking that involves a team of people using the tester in various capacities:

- ***Test plan developers*** work with the design engineer to translate part specifications and operating limits into a set of measurement on the tester to determine the functionality of parts.
- ***Test floor managers*** implement the tester setups and procedures to support these test, supervising the logistics associated with the production environment.
- ***Operators*** execute the designed tests quickly, maintaining tester uptime and throughput of parts.

Each responsibility is crucial to successful production test and in order to best support the particular capacities of each type, the Cassini software can be adapted to suit their specific needs. This is accomplished by establishing user accounts on the system, creating a separate work environment for each person, and initiating another advanced feature of the system: ***user roles***. Within Cassini there are four types of roles :

Debug	**Engineer**	**Maintenance**	**Operator**
For RI Use Only	**Highest Ranking User**	**2nd Ranked User**	**3rd Ranked User**
Access to underlying code for debug and upgrading feature sets	Unlimited access to entire test systems: test plans, Fixture definitions, DIB configurations, etc.	Access to system calibration and hardware debug information.	Limited access to only the test exec to run production test

Software Access Control



The above diagram shows Cassini's system permission topology, comprised of two main control mechanisms: user roles and groups. Both exploit the tester's object-oriented software architecture to set application access and manage services:

- *User Roles* allow the system to augment the user interface and limit what a user can see in menus
- *Groups* provide a user-defined mechanism to limit access to system applications, data, and resources

Delegating Responsibilities

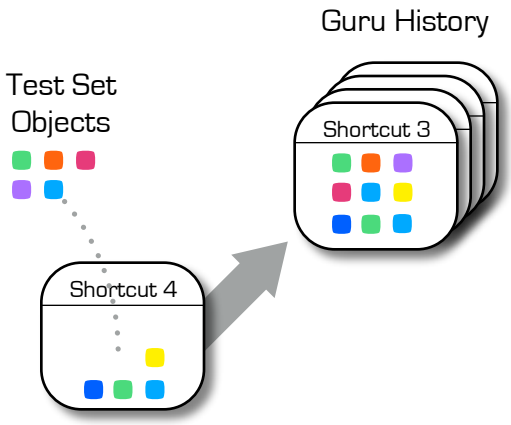
Depending on the rank of the user, the system links a set of corresponding permissions to access applications within the tester. The higher the rank, the more applications the user can access. If a floor operator was going to run a production test for instance, an engineer does not want that operator to be able to directly or inadvertently change a test plan or alter the calibration information of the tester. By assigning that user to an operator role, the system will not only restrict what files and data can be changed, but limit the interface to show only commands that execute a set test plan. If a higher-ranking user wanted to give specific access to certain parts of the tester to a lower-ranking user, they can assign a collection of custom permissions by creating groups and linking those users to a particular group or set of groups. This delegated access control was designed to be highly flexible, and can have overlapping and isolated permissions to support multiple users with custom or complex access needs. For instance, if an engineer user wanted to allow a maintenance user to have access to data within a test plan, or a subset of data within the test plan for calibration purposes, a group with these permissions could be used. Another common scenario would be to use groups to manage separate projects on the tester that involve multiple engineer users. This allows each to access applications and files associated with their project, while hiding and barring access to other engineer's projects on the test system. This gives each individual a simple, clean, and familiar interface on the tester to manage data and perform their tasks efficiently without the fear of altering another user's setup or files.

Development Sandbox

In addition to the separation of each user's workspace from one another, Guru enables a user to isolate design spaces using a "digital sandbox" within their work environment. When software patches with updated firmware and/or advanced features are released, the user can experiment with them against their current hardware/software configuration using a sandbox. By marking the software objects in the patch with the sandbox setting, the contents are kept isolated from the rest of the user's software and design setup. This provides a software queue where the developer can control the release of new feature sets to the tester at their discretion.

Shortcut Hygiene

Much like the development sandbox, Guru provides the ability to quickly and easily save and roll back test setups and software configurations to a previous setting, giving the user the freedom to experiment without the fear of losing a previous design iteration. To make the step of going back to a previous setup easier, the software allows the user to save an entire configuration as a shortcut. Using the Guru History interface, the user can sort through a chronological record of shortcuts, and then select one to resort back to a previous setup. By taking the time to create shortcuts, the user always has a quick and simple way of recovering their setup.



Typical Development Workflow

Once permissions and system access have been set, it's time to build a test. Cassini's workflow has a device-oriented workflow that starts at the part and works backward to the tester and into the design of the measurements in software:

1. Assign resources to the device pins.

- This is the development of the physical interconnect between the tester and the device under test.
- Helps to determine the design of the device interface board (DIB) and configurations of the Fixture and TIMs.

2. Create a software Fixture and device interface.

- The physical connections between the tester's Fixture and DUT are defined in the software so that the system can automate the switching of signal needs to support measurements.

3. Define and create a software device.

- The communication link to manipulate and set control logic in the DUT is established to allow the tester to automate the control setup of the part.
- Interconnect paths from the test instrument modules (TIMs) to the Fixture are defined.

4. Create a test plan

- Now that all of the physical interconnects are in place, and the tester has defined control and signal paths to the DUT, the user can build a test program to validate the part.

In the background, Guru organizes and recalls these definitions, test setups, and test resource configurations according to the user's meta tags, to keep a running log of design iterations and current tests. When it is time to release a test to the production floor, Guru provides the distribution gateway to pass the test plans to other testers.

Getting Help

Cassini has multiple layers of help to assist a user in the development process. Within the test plan software, each function and command has a linked help guide that provides information on the functionality and use of the particular object block. As well, the RI website contains an extensive knowledge base of: practical examples of test plans, calibration info, FixtureFixture basics, startup guides, application notes, and trouble shooting guides. In addition to the resources available to the user, Guru also serves as a help request resource. Test plans, setups, and the user's Guru configuration files can be compressed into a zip file and sent to RI through Guru's network hub. This provides a quick and easy way for RI engineers to analyze customer issues by allowing them to literally "peek under the hood" of their software setup remotely to help provide support.

Summary

This concludes the Cassini design philosophy and introductory material. For more information please visit our website: http://www.roos.com
or contact us at:

Roos Instruments
2285 Martin Ave
Santa Clara, CA 95050
(408)748-8589
sales@roos.com