Script Information

Type	Article
SUN #	S4284
Title	Article: Test Drivers and Stubs
Keywords	
Script Text	

Introduction

Software testing requires cooperation between the testing group and the software developers. This article discusses how developers can help prepare some parts of the software to assist in the testing process.

Test Drivers

Many software packages are designed to operate or use other devices. In many cases, these devices are not available during the development process, but the interfaces to these devices must be tested.

The software interface to an external device is called a driver. Drivers encapsulate the complex communication with a device into simple program interfaces that developers can use in their programs. For example, a device may be designed to collect weather information such as temperature and humidity. This device will have an associated driver with programming interfaces such as reset, read data, etc.

If the device is not available when the software is developed, a test driver can be prepared to simulate the operation of the device. In many cases, another computer is used to serve as the device, and a test driver is written to communicate with the service computer in ways that will simulate the target device. Developers know the expected operation of the target device and can prepare the test driver to act accordingly. Using this technique, the main application can be prepared and executed against the test driver with all programming interfaces tested in absence of the external device.

Stubs

Software is typically developed in pieces or modules. As functionality is added to the software, relationships and dependencies begin to form. Module A is designed to rely on module B and Module C. In the development plan, Modules B and C may not yet be developed when Module A is written. How do the developers proceed, and how is module testing performed when dependencies in the code are not yet completed?

Stubs are similar to test drivers in that they simulate the operation of a required programming resource. In the example above, assume Module B is required to perform a complex database query and return the results. Module A is the caller and cannot be written and tested without Module B. The solution is to write

Module B as a stub which returns expected results to the caller but is written in a simplistic form, perhaps with hard-coded data. The goal is to make sure Module B returns expected results without having to go through the effort of full implementation out of sequence with the project plan.

When software designs are complex, the use of stubs is critical to avoid the deadlocks that would occur because of the interdependencies of the code. Preparation of stubs also allows the testers to proceed with module testing and reduces the reliance of the test team on the development team. The popularity of object-oriented programming practices makes the design of stubs easier in some respects because of the ability to use base class methods for stubs.

Summary

This article summarized some of the ways software developers can prepare software to assist in the testing process. Test drivers can be used to emulate the operation of drivers or external resources that are not yet in place, and stubs allow programs to be built before all the code is complete.