

## The Test Executive: The Strength of Testing and Diagnosis

Calvin Carlson WesTest Engineering 810 W. Shepard Ln Farmington, Utah 84025 801-451-9191 ext 126 email: <a href="mailto:Cal@westest.com">Cal@westest.com</a>	Larry V. Kirkland WesTest Engineering 810 W. Shepard Ln Farmington, Utah 84025 801-451-9191 ext 124 Email: <a href="mailto:LarryK@westest.com">LarryK@westest.com</a>
---	--

*Abstract - The Test Executive's primary purpose is to control the flow of testing on a given piece of automated test equipment. It accomplishes this by loading and executing a test program's compiled executable code. Code execution enables the test executive to control the instruments, make decisions and direct the testing in accordance with the instructions contained in the test program. In addition to this primary basic function; Test Executives can be enhanced to provide additional options to improve overall testing and diagnosis.*

*Debug tools can also be built into the Test Executive in order to provide a rich set of debug options for the test engineer. These options should be readily useable and easy to perceive. Also, since diagnosis is a huge part of Unit Under Test (UUT) testing, the debug capabilities should be vast and readily utilizable.*

*In addition, backward compatibility, built-in securities, ease of operator use, reliability, and maintainability should also be considered in order to produce the ultimate Test Executive. Keeping this in mind, a team of experienced test engineers designed and developed a Test Executive (known as the WT2000) rich with execution options, debug tools, development tools and other vital TPS aids the overall performance of a*

*strong Test Executive as a result of these enhancements and will be discussed in detail in this paper.*

*The vastness or strength of debug tools and will also be discussed in this paper. This paper doesn't merely identify options and user friendliness of a Test Executive, but also the paper explains a real application and an actual test environment and the various techniques an operator and an engineer can utilize to perform test and diagnosis.*

*The paper will also explain the use of a Test Station Emulator and how it allows test programs to be run and debugged using work stations not connected to the test system. Also, shown and explained will be in-depth debug capabilities and Soft Front Panels that allow instruments to be controlled from the test system's computer console.*

*Another prime area which will be explained is a Test Navigator that provides the test station with its operators interface. This is a Browser Based Interface, i.e., acting much like an Internet Browser. This makes the test station extremely easy to operate and reduces training requirements to an absolute minimum.*

**Keywords** – *Test Executive, test program set, diagnosis, operator interface, debug tools, emulation, WT2000*

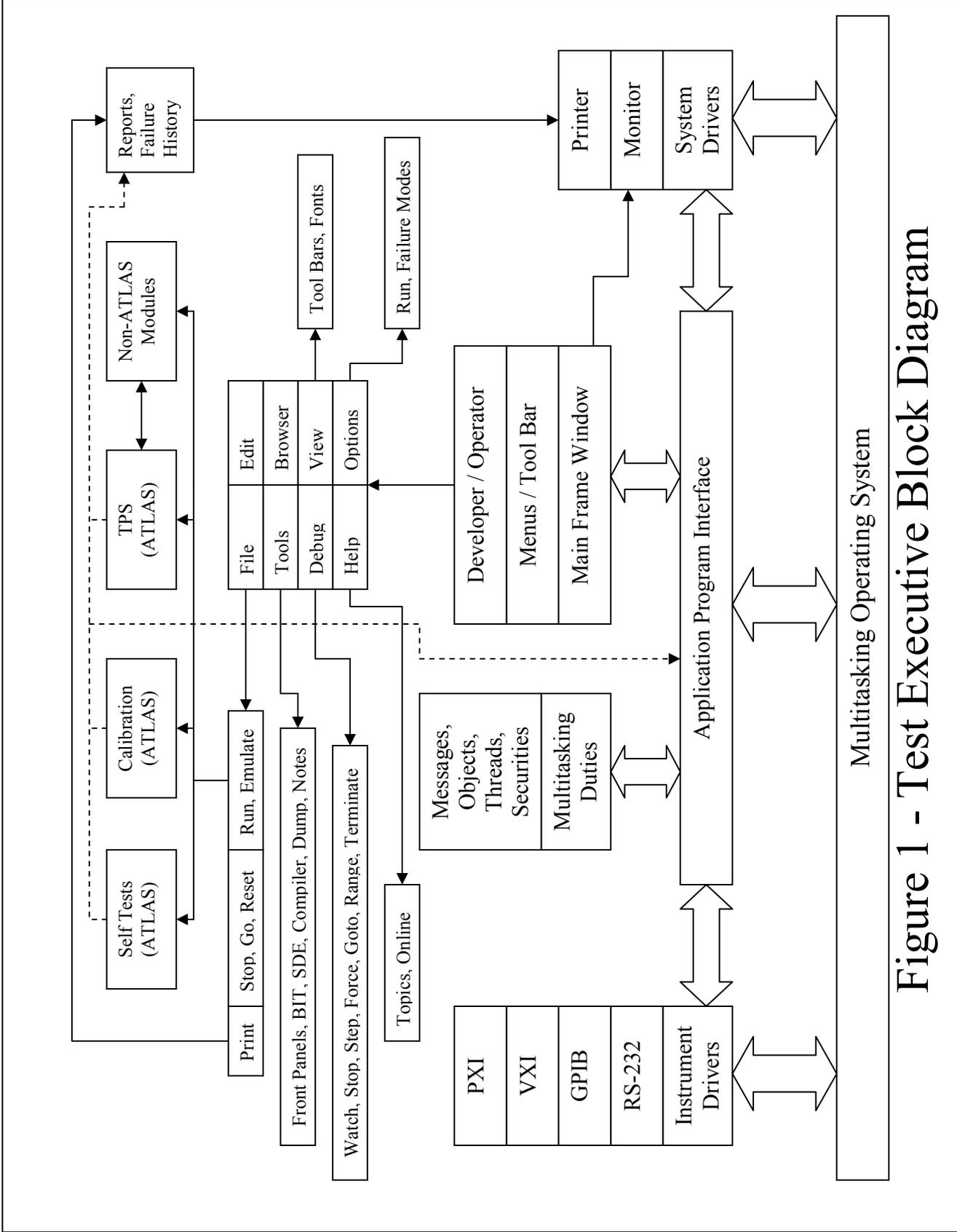


Figure 1 - Test Executive Block Diagram

## I. INTRODUCTION

The purpose of this paper is to explain how the overall performance of a test system's testing and diagnostic capabilities depend on the Test Executive used to process the test program known as a Test Program Set (TPS).

With commercial of the shelf (COTS) test equipment readily available, assembling an assortment of VXI, PXI, LXI or GPIB hardware to satisfy a wide range of testing capabilities is not too difficult. However, getting the hardware to perform a wide variety of automated tasks in the form of a TPS relies heavily on the system software. System software includes the operating system (OS), the Test Executive, instrument drivers and associated dynamic link libraries (DLL), and the TPS executables. While all are important, the piece of software that pulls the others pieces together to create a high performance test environment is the Test Executive. The block diagram shown in Figure 1 helps portray how the software pieces interact and work with each other.

From the diagram you can see that the Test Executive takes advantage of a multitasking, multithreaded operating system in order to maximize execution performance. The diagram also illustrates other details that will be discussed later. It also helps illustrate how and why the overall strength, performance and efficiency of a test system rely on the Test Executive.

This paper is divided into three additional sections; the first section explains the Test Executive's real time applications, the second details the various offline capabilities, and the third section provides the summary and conclusion.

## II. REAL TIME APPLICATIONS

It is during TPS execution that real time Test Executive tools are implemented. You will note that ATLAS is the preferred TPS format for the Test Executive discussed in this paper, however, ATLAS programming will not be addressed in any detail in this paper. Real time tools include execution options, integration and development tools, and the operator interface.

### A. Execution Options

Key to test execution is the mode in which test programs can execute in a test system. The life cycle of a TPS involves a number of operators which include developers, integrators, quality and end users as a minimum. Implementing securities into the Test Executive allows operators to only run in the mode they have been given privileges for. The four modes defined by the Test Executive discussed in this paper (WT2000) are normal, development, integration and qualification and are determined by the log in set up by the system administrator. Each operator will need to be assigned to at least one user's group in order to have privileges for test system use. The execution mode will also determine operator entry (operator name, serial numbers, fault inserted if in integration mode, etc..) and the data that is recorded during test execution.

Another real time option that is used specifically for digital test diagnostics is the selection of the guided probe and or fault dictionary. Having the capability to select one or the other or both is often crucial in fault diagnosis.

Other options made available by the WT2000 to the test operator include the ability to enable a history failure data base or enable synthesized speech during probe sequences

## B. Integration and Development Tools

To merely run a test program and not provide the engineer and test technician with tools to debug a test program or analyze test results is of little value in a true test/diagnosis environment. Some still think ATE Test Programs are flawless and know all and can tell all. In fact most test programs are not robust enough to perform all the required diagnosis to properly fault isolate. Tools which allow the skill of the technician to analyze test data and control the flow of testing are absolutely critical. Indeed, substantial debug options are so critical to proper test/diagnosis that they should be mandated in all Test Executives.

Integration and development tools implemented in the WT2000 include run time commands, specific debug commands, a soft reset, system program execution and Soft Front Panels. All of these tools are very intuitive and can be executed by a drop down menu command or by selecting the corresponding icon on the tool bar as shown in Figure 2.

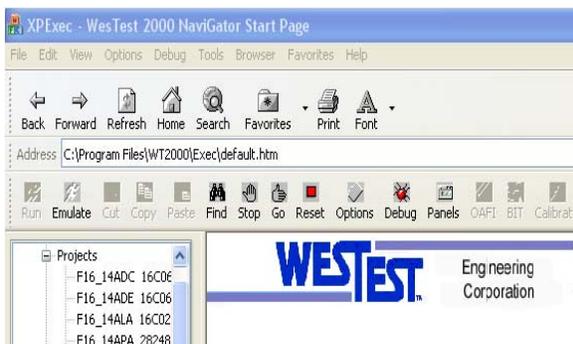


Figure 2 – Menu and Tool Bar

The stop and go commands provide the operator the ability to stop and go at any time during test program execution. This is valuable when specific stop conditions are not known to the test program operator and can be invoked with a single touch of the escape key.

Debug tools contribute more to the strength of a Test Executive than any thing else. The debug tools provided with the WT2000 have been implemented with the developer and test program integrator in mind.. They are very useful, intuitive and easy to use. Securities have also been implemented for these tools in order to restrict access and provide protection for the test system and UUT. The Debug commands implemented in the WT2000 include the following:

- Watch window for global variables
- Stop on test number
- Stop on variable value
- Stop on Nogo
- Single step
- Force test to Go
- Force test to Nogo
- Go to test
- Run test range
- Terminate tests

A soft reset has been provided by the Test Executive and is treated as if the hard reset button were pushed. This feature provides additional safety to the test system.

The capability to perform test system analysis is often desired to test the integrity of the test system. Test System analysis applications include the system self test, system calibration and system BIT for all instruments. Having the capability to execute these specific system tests within the Test Executive enhances system reliability and maintainability.

Being able to quickly pull up an instrument's Soft Front Panel within the Test Executive gives the operator quick access for monitoring a particular test or changing an instruments configuration or state. Once again, securities will be in affect and only certain users will have full capability to all instruments and their features.

### C. Operator Interface

Due to the wide spread use of the World Wide Web, the interface used in the WT2000 known as the Test Navigator is browser based and has the ability to perform basic browser functions such as moving forward and back, refreshing, accessing favorites, changing font size and so forth. This makes operation of the WT2000 very easy and intuitive. Reports generated during test program execution are stored in HTML format and can be readily printed or archived by the test operator. The main page has the ability to scroll up and down and is continually updated by the output of the test program. Other visuals provided by the Test Executive include the tool bars, status bar, control bar and debug bars. The status bar keeps the operator informed of test numbers, measurements, limits, units and go or nogo status during TPS execution. Operator intervention required during test execution is accomplished with a variety of dialog boxes (static or dynamic) presented to the operator.

### III. OFFLINE TOOLS

Probably one of the biggest strengths of the WT2000 is the capability to emulate a TPS offline. Not only is it critical for the test developer to have the capability to develop and compile test programs offline, but to execute them as well. This is done by invoking an

emulation mode that simulates instrument writes and reads and enables the executable code to run as if it were being run on the test station. This in turn enables the developer to test a big portion of his code without encroaching on valuable test system time. Keep in mind that all debug tools are still in effect, so programs can be forced, single stepped, watched and so forth.

Other offline features include the ability to have direct access to the ATLAS compiler and other applications used for TPS development.

Online help and documentation provides the operator a vast library of information at their fingertips and has been proven to be very valuable during TPS development.

### IV. SUMMARY AND CONCLUSION

There is no question that the overall test performance and strength of a test system lays within the Test Executive. This is the case whether the Test Executive has a lot of features that aid the test program developer or not. However, we know that providing a powerful Test Executive full of debug capabilities and integration tools is what is needed for efficiency and high level performance in the world of automated test equipment. Without a doubt, such a Test Executive has a big impact on diagnostics, reliability and maintainability. It is also important to understand that backward compatibility, built-in securities and ease of operator use is also essential for a Test Executive to be complete and provide ultimate support for the test program developer.