



Q'n Apps #E12

Q: How can I best organize my small production runs without an IC handler?

ETS2k for Manual Test Applications - The HILEVEL Test Box

While the application excludes automatic handling equipment, there nevertheless is a striking resemblance between tester applications employing human handlers of devices and applications that utilize robots for moving parts on the basis of the outcome of the test. Indeed, the screens pertaining to the TestBox are a subset of the screens that cover IC Handler utilization.

Before either of the aforementioned screens can be utilized, we must build the test using Build ATX in the AutoTest sidebar. There we generate our test by picking commands from a menu. Some of these commands, such as DUT CONTINUITY and FUNCTIONAL are self-contained provided you associate the proper SET file and TRN (Vector) file with the command. Other commands may require further association. For instance, DC PMU tests such VOH test must be associated with a PMU test defined for the particular set file. This test will be attached to the command using the **Attach** button.

In this way you can structure a complex **TestSuite**, applying multiple sub-tests associated with multiple SET files and Vector files. When done, you save the file with the default extension ATX and exit the Build ATX screen. We presume you maintained the default **QuickLoad** that lets you utilize the tester vector memory as a cache for all the vector files you are using for the test (if your memory modules are not sufficiently large to hold all the vector files, you will need to uncheck the **QuickLoad** option).

Upon entering the **TestBox** window, we load the ATX file, which loads all vector files into the tester memory. Sub-tests using the same vector files will share the vector space in the tester memory. The obvious benefit of this mechanism is that we don't need to download vector files from disk during execution of the test, resulting in extremely fast deployment of multiple vector files.

You next press **Start** and you are ready to run the test by pressing the button on the **TestBox**. Each sub-test will indicate pass/fail, and the final outcome of the test will be celebrated by a big green smiling face or bemoaned by a big red sad face.

The time (with millisecond resolution and the accuracy of a crystal oscillator) taken to execute each sub-test is indicated adjacent to each check mark.

Branching is also supported. For instance, if a test fails, it may take an alternate path to further ascertain the type of failure. Or, in the case of failing continuity, you may skip the rest of the tests and go straight to END; there is typically no need to run the rest of the tests if the part is not connected.

Also See:

Q'nApp #E3: HILEVEL Production Floor

Q'nApp #E11: The TestBox

Q'nApp #E14: Production Test

Q'nApp #E18: Prober Fixtures

Q'nApp #E37: MultiSite

Q'nApp #E48: Custom AutoTest Logfile

Q'nApp #E51: Fast Continuity