**[Static Versus Dynamic Testing Tools](http://www.softwaretestinggenius.com/articals.php?mode=details&qry=635&parent=84)**

Since testing is of two types like 1) Static testing 2) Dynamic testing; accordingly the tools used during these testing are also known as

1) Static testing tools

2) Dynamic testing tools

Static testing tools seek to support the static testing process whereas dynamic testing tools support dynamic testing process. It may be noted that static testing is different from dynamic testing.   
  
**Few points of differences among static and dynamic testing are as under:**

|  |  |  |
| --- | --- | --- |
|  | **Static Testing** | **Dynamic Testing** |
| 1 | Static testing does not require the actual execution of software. | Dynamic testing involves testing the software by actually executing it. |
| 2 | It is more cost effective. | It is less cost effective. |
| 3 | It may achieve 100% statement coverage in relatively short time. | It achieves less than 50% statement coverage because it finds bugs only in part of codes those are actually executed. |
| 4 | It usually takes shorter time. | It may involve running several test cases, each of which may take longer then compilation. |
| 5 | It may uncover variety of bugs. | It uncovers limited type of bugs that are explorable through execution. |
| 6 | It can be done before compilation. | It can take place only after executables are ready |

Software testing tools are frequently used to ensure consistency, thoroughness and efficiency in testing software products and to fulfil the requirements of planned testing activities. These tools may facilitate unit (module) testing and subsequent integration testing (e.g., drivers and stubs) as well as commercial software testing tools.

**Testing tools can be classified into following two categories:**

**Static Test Tools:** These tools do not involve actual input and output. Rather, they take a symbolic approach to testing, i.e. they do not test the actual execution of the software. These tools include the following: ,

**1) Flow analyzers:** They ensure consistency in data flow from input to output.

**2) Path tests:** They find unused code and code with contradictions.

**3) Coverage analyzers:** It ensures that all logic paths are tested.

**4) Interface analyzers:** It examines the effects of passing variables and data between modules.

**Dynamic Test Tools:** These tools test the software system with 'live' data. Dynamic test tools include the following

**1) Test driver:** It inputs data into a module-under-test (MUT).

**2) Test beds:** It simultaneously displays source code along with the program under execution.

**3) Emulators:** The response facilities are used to emulate parts of the system not yet developed.

**4) Mutation analyzers:** The errors are deliberately 'fed' into the code in order to test fault tolerance of the system.