

## Test Data Generation: What is, How to, Example, Tools

As a tester, you may think that 'Designing Test cases is challenging enough, then why bother about something as trivial as Test Data'. The purpose of this tutorial is to introduce you to Test Data, its importance and give practical tips and tricks to generate test data quickly. So, let's Begin!

### What is Test Data in Software Testing?

**Test Data in Software Testing** is the input given to a software program during test execution. It represents data that affects or affected by software execution while testing. Test data is used for both positive testing to verify that functions produce expected results for given inputs and for negative testing to test software ability to handle unusual, exceptional or unexpected inputs.

Poorly designed testing data may not test all possible test scenarios which will hamper the quality of the software.



### What is Test Data Generation? Why test data should be created before test execution?

Everybody knows that testing is a process that produces and consumes large amounts of data. Data used in testing describes the initial conditions for a test and represents the medium through which the tester influences the software. It is a crucial part of most [Functional Tests](#).

Depending on your testing environment you may need to CREATE Test Data (Most of the times) or at least identify a suitable test data for your test cases (is the test data is already created).

Typically test data is created in-sync with the test case it is intended to be used for.

Test Data can be Generated -

- Manually
- Mass copy of data from production to testing environment
- Mass copy of test data from legacy client systems
- Automated Test Data Generation Tools

Typically sample data should be generated before you begin test execution because it is difficult to handle test data management otherwise. Since in **many testing environments creating test data takes multiple pre-steps or very time-consuming test environment configurations**. . Also If test data generation is done *while* you are in test execution phase you may exceed your testing deadline.

Below are described several testing types together with some suggestions regarding their testing data needs.

### Test Data for White Box Testing

In [White Box Testing](#), test data Management is derived from direct examination of the code to be tested. Test data may be selected by taking into account the following things:

- It is desirable to cover as many branches as possible; testing data can be generated such that all branches in the program source code are tested at least once
- Path testing: all paths in the program source code are tested at least once - test data preparation can done to cover as many cases as possible
- Negative [API Testing](#):
  - Testing data may contain invalid parameter types used to call different methods
  - Testing data may consist in invalid combinations of arguments which are used to call the program's methods

### Test Data for Performance Testing

[Performance Testing](#) is the type of testing which is performed in order to determine how fast system responds under a particular workload. The goal of this type of testing is not to find bugs, but to eliminate bottlenecks. An important aspect of Performance Testing is that the set of sample data used must be very close to *'real' or 'live'* data which is used on production. The following question arises: 'Ok, it's good to test with real data, but how do I obtain this data?' The answer is pretty straightforward: from the people who know the best – **the customers**. They may be able to provide some data they already have or, if they don't have an existing set of data, they may help you by giving feedback regarding how the real-world data might look like. In case you are in a **maintenance testing** project you could copy data from the production environment into the testing bed. It is a good practice to **anonymize** (scramble) sensitive customer data like Social Security Number, Credit Card Numbers, Bank Details etc. while the copy is made.

### Test Data for Security Testing

[Security Testing](#) is the process that determines if an information system protects data from malicious intent. The set of data that need to be designed in order to fully test a software security must cover the following topics:

- **Confidentiality:** All the information provided by clients is held in the strictest confidence and is not shared with any outside parties. As a short example, if an application uses SSL, you can design a set of test data which verifies that the encryption is done correctly.
- **Integrity:** Determine that the information provided by the system is correct. To design suitable test data you can start by taking an in-depth look at the design, code, databases and file structures.
- **Authentication:** Represents the process of establishing the identity of a user. Testing data can be designed as a different combination of usernames and passwords and its purpose is to check that only the authorized people are able to access the software system.
- **Authorization:** Tells what are the rights of a specific user. Testing data may contain a different combination of users, roles and *operations* in order to check only users with sufficient privileges are able to perform a particular operation.

### Test Data for Black Box Testing

In Black Box Testing the code is not visible to the tester. Your functional test cases can have test data meeting following criteria -

- **No data:** Check system response when no data is submitted
- **Valid data:** Check system response when Valid test data is submitted
- **Invalid data:** Check system response when *Invalid* test data is submitted
- **Illegal data format:** Check system response when test data is in an invalid format
- **Boundary Condition Dataset:** Test data meeting boundary value conditions
- **Equivalence Partition Data Set:** Test data qualifying your equivalence partitions.
- **Decision Table Data Set:** Test data qualifying your decision table testing strategy
- **State Transition Test Data Set:** Test data meeting your state transition testing strategy
- **Use Case Test Data:** Test Data in-sync with your use cases.

**Note:** Depending on the software application to be tested, you may use some or all of the above test data creation

### Automated Test Data Generation Tools

In order to generate various sets of data, you can use a gamut of automated test data generation tools. Below are some examples of such tools:


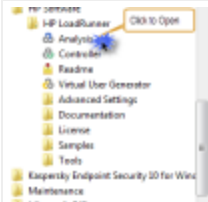




DTM [Test Data generator](#), is a fully customizable utility that generates data, tables (views, procedures etc) for database testing (performance testing, QA testing, load testing or usability testing) purposes. Datatect is a [SQL](#) data generator by Banner Software, generates a variety of realistic test data in ASCII flat files or directly generates test data for RDBMS including Oracle, Sybase, SQL Server, and Informix.

### Conclusion

In conclusion, well-designed testing data allows you to identify and correct serious flaws in functionality. Choice of test data selected must be reevaluated in every phase of a multi-phase product development cycle. So, always keep an eye on it.

[< Prev](#)   [Report a Bug](#)   [Next >](#)

#### YOU MIGHT LIKE:

<div>SOFTWARE TESTING</div> <div></div> <div><a href="#">Difference Between Retesting and Regression Testing</a></div> <div>Retesting Retesting is a process to check specific test cases that are found with bug/s in the...</div> <div><a href="#">Read more »</a></div>	<div>LOADRUNNER</div> <div></div> <div><a href="#">LoadRunner Analysis Tutorial: Report &amp; Graph</a></div> <div>In this tutorial, we will study following topics- 1. How to use Analysis in LoadRunner</div> <div>2....</div> <div><a href="#">Read more »</a></div>	<div>SOFTWARE TESTING</div> <div></div> <div><a href="#">What is Scalability Testing? Learn with Example</a></div> <div>Scalability Testing Scalability Testing is a non functional testing method that</div> <div>measures...</div> <div><a href="#">Read more »</a></div>
<div>ENTERPRISE TESTING</div> <div></div> <div><a href="#">Database (Data) Testing Tutorial with Sample Test Cases</a></div> <div>What is Database Testing? Database Testing is a type of software testing that checks the schema,...</div> <div><a href="#">Read more »</a></div>	<div>SOFTWARE TESTING</div> <div></div> <div><a href="#">HealthCare Domain Testing with Sample Test Cases</a></div> <div>Before we begin testing, let's quickly study the basic healthcare domain</div> <div>knowledge. HealthCare...</div> <div><a href="#">Read more »</a></div>	<div>COURSE</div> <div></div> <div><a href="#">Free Mobile App Testing Tutorial</a></div> <div>Training Summary Android &amp; iOS are the most popular mobile OS. There are millions of application designed...</div> <div><a href="#">Read more »</a></div>

### Top Tutorials



#### About

[About Us](#)  
[Advertise with Us](#)  
[Write For Us](#)  
[Contact Us](#)

#### Career Suggestion

[SAP Career Suggestion Tool](#)  
[Software Testing as a Career](#)

#### Interesting

[eBook](#)  
[Blog](#)  
[Quiz](#)  
[SAP eBook](#)

#### Execute online

[Execute Java Online](#)  
[Execute Javascript](#)  
[Execute HTML](#)  
[Execute Python](#)



Selenium



Testing



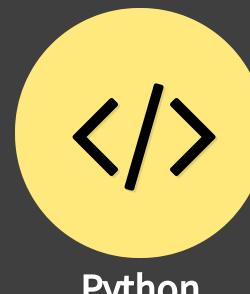
Hacking



SAP



Java



Python



Jmeter



Informatica



JIRA