

Approaches to Test Data

Mike Kelly

First Published on SQAtester.com

What is Test Data?

Test data is, for the purposes of this article, that which is entered into the application under test for verification purposes or is used to drive the application by executing processes for assisting in automated navigation. Specifically, we will be talking about test data as it applies to automated testing. This article will cover two different approaches to test data that will make your data more robust, easier to maintain, and should reduce long-term costs.

Reverse Delta Processing

Reverse delta processing is a method of implementing dates within your test data so that they never age. Take for example the following screen shot of an application used to process credit card applications.

Personal Information			
First Name	M.I.	Last Name	Sfx
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Social Security #	Birthdate (mm/dd/yyyy) ¹		Mother's Maiden Name
<input type="text"/>	<input type="text"/>		<input type="text"/>
Email Address	Yearly Household Income ²		
<input type="text"/>	\$ <input type="text"/> .00 (Gross Annual)		
Address Information			
Street #	Street Name	Apartment #	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
City	State	Zip Code	
<input type="text"/>	<input type="text"/>	<input type="text"/>	
Home Phone	Residential Status		
(<input type="text"/>) - <input type="text"/>	<input type="text"/>		
Employment Information			
Employer			
<input type="text"/>			
Employer Phone	Employed There		
(<input type="text"/>) - <input type="text"/>	<input type="text"/> years, <input type="text"/> months		

If you were to develop tests for this application one of the things you would test would be that the person who is applying for the credit card is of legal age to do so. So in an effort to develop a set of tests that exercise the boundaries of the Birthdate field you could come up with the following table of possible test dates:

Value	Result	Comment
04/02/1985	Approved	18 yrs old
04/02/1986	Denied	17 yrs old

If you develop regression tests based on this data then (all other things being equal) your tests should run correctly until the date 04/02/2004. At this time your second test (BD = 04/02/1986) will be approved and you will have to go back, figure out why the application was approved, and change the test data. Further, test one (DB = 04/02/1985) is no longer a good test either because it no longer tests the boundaries of the algorithm.

The reverse delta concept works off of the current date and adjusts it according to a positive or negative date value. Again by example, if you had the following table:

Reverse Delta	Result	Comment
-000000018	Approved	18 yrs old
-000000017	Denied	17 yrs old

What happens in this scenario is you have a function that processes the reverse delta value and enters the dates 04/02/1985 and 04/02/1986 respectively. The next year this test is run it will enter 04/02/1986 and 04/02/1987 respectively. Your dates do not age according to the current date. All dates are a function of the current date.

You can use any format you like for your implementation of reverse delta, but I prefer the following: \pm DDDDMMYYYY; where D = day, M = month, and Y= year.

Example	Result
+100000000	One hundred days in the future.
-000020000	Two months ago.
+000000001	One year in the future.
-003020001	One year, two months, and three days ago.

I will leave the mechanics of the reverse delta function to you as your platform, language, and specific implementation will more than likely be unique. It should simply be a read from a test input, process the reverse delta value, and return the result for population in the application-under-test.

Default Data vs. Test Data

By setting up a default dataset you can save space and time on the creation of test data as well as simplify the test dataset making it easier to read and problem shoot. Using the same example of the credit card application lets look at some test data for the Address Information section:

Address Information

Street #

Street Name

Apartment #

City

State

Zip Code

Home Phone

Residential Status

(

)-

Lets assume we are in New York and we want to test the processing of out of state applications. Our test data may look something like this:

Street #	Street Name	Apart #	City	State	Zip Code	Phone	Status
1234	Washington Blvd		Penn Yan	NY	14527	555-345-4776	Owner
1234	Washington Blvd		Berne	IN	46711	555-345-4776	Owner
1234	Washington Blvd		Big Lake	TX	76932	555-345-4776	Owner
1234	Washington Blvd		Yakima	WA	98901	555-345-4776	Owner
1234	Washington Blvd		Wellston	OH	45692	555-345-4776	Owner

If I were to break this apart into a default and a test dataset I would have the following:

Default Data:

Street #	Street Name	Apart #	City	State	Zip Code	Phone	Status
1234	Washington Blvd		Penn Yan	NY	14527	555-345-4776	Owner

Test Data:

Street #	Street Name	Apart #	City	State	Zip Code	Phone	Status
			Berne	IN	46711		
			Big Lake	TX	76932		
			Yakima	WA	98901		
			Wellston	OH	45692		

Looking at the tables above you can very quickly identify the test variables for the test dataset. One can quickly determine what is being tested and what values are used. If this were a much larger table (all states, more cities per state, etc...) you would more readily appreciate the simplifications.

When implementing this model you will need a process that reads from both the test and the default dataset. That process should populate data using the default data whenever it encounters an empty cell (or null value) in the test data set. The very simple illustration below demonstrates this process.

