



Upgrade Path Testing



A Kanbay Whitepaper

By Dhiraj Lokhande



WWW.KANBAY.COM

©2007 Kanbay Incorporated. All rights reserved. Kanbay and the Kanbay logo are registered trademarks of Kanbay Incorporated. Other trademarks are the property of their respective owners.



Abstract

Upgrade path testing, also called up-gradation testing, sounds a bit unusual. But testing upgrades is an important part of the software life cycle.

What exactly does upgrade path mean to software? Suppose your company developed a product which was rolled out into the market with version 1.0. Due to few showstopper defects, your company needs to deliver version 1.1 which could be a Patch or a Hot Fix. What are the different things which should be evaluated and tested so that we can say the up-gradation is working correctly? How do we test an upgrade?

On a high level, we test an upgrade path using the following techniques:

- » Installation Testing
- » Database Testing
- » Application Testing
- » Documentation Testing

Installation Testing

Why is this type of testing required?

When you are delivering a new version with bug fixes there is a high possibility that the code is changed or modified. Installation testing helps confirm that the right changes are going into the correct folders. For example, suppose there is a folder which contains all the HTML files which have updated. Does the CSS folder also need to be updated? If we update the HTML files without updating related CSS files, the color/fonts and other attributes will not be reflected correctly.

How do we test?

Most of the time, installation testing is done with the help of the Software Configuration Management (SCM) team as we have to evaluate both the old system and the upgraded path.

The first testing is **folder level validation**. All folders with the timestamps are checked so we can revisit which are the folders that are copied to the system by this installation procedure.

Second the size of the files which are modified are tested and compared with the earlier version. This could be white or black box testing. In white box testing, the tester must open the file and check if the code actually exists or not. In black box testing the tester must check the file size since it should definitely differ from the old file size. If the tester finds any discrepancies he or she can open a defect with HIGH severity and HIGH priority.

Some companies perform installation testing for the complete application. For example, suppose a new company would like to buy version 1.1 directly and is not concerned about

Upgrade Path Testing

the issues or settings for V1.0. Installation testing must be performed for this scenario in a similar manner.

The diagram below shows the verification of the installation testing process. New code is copied into a folder "patch" and the tester views the change in the timestamp of the folder.

Exhibit 1: Verification Process of Installation Testing

The exhibit consists of two side-by-side command-line windows, both titled "C:\WINNT\system32\cmd.exe".

Before the installation: This window shows a directory listing from January 18, 2007, at 11:29 AM. A specific file, "load_trans.sql", is highlighted with a green oval. Its timestamp is 09/08/2006 05:59 PM. The file size is 1,075 bytes.

Date	Time	Type	Name	Size
01/18/2007	11:29 AM	<DIR>	.	
01/18/2007	11:29 AM	<DIR>	..	
09/08/2006	04:01 PM		3,645 apexins.sql	
07/18/2006	06:12 PM		6,698 apexvalidate.sql	
09/08/2006	04:01 PM		9,856 apex_egg_config.sql	
09/21/2006	03:51 PM		3,439 apxdbnig.sql	
08/21/2006	03:51 PM		1,537 apxremov.sql	
07/24/2006	03:59 PM		2,877 apxxenig.sql	
09/08/2006	04:08 PM		1,261 apxxepwd.sql	
01/18/2007	11:29 AM	<DIR>	builder	
09/08/2006	04:02 PM		3,972 catapx.sql	
01/18/2007	11:29 AM	<DIR>	core	
09/08/2006	04:06 PM		85,975 coreins.sql	
09/08/2006	04:02 PM		86,872 coreins2.sql	
01/18/2007	11:28 AM	<DIR>	doc	
01/18/2007	11:29 AM	<DIR>	images	
09/08/2006	05:59 PM		1,075 load_trans.sql	
01/18/2007	11:28 AM	<DIR>	patch	
01/18/2007	11:29 AM	<DIR>	utilities	
10/10/2006	07:07 AM		3,985 welcome.html	
			12 File(s) 211,192 bytes	
			8 Dir(s) 22,976,638,976 bytes free	

After the installation: This window shows the same directory listing after the installation. The "load_trans.sql" file has been moved to the "patch" directory and its timestamp has changed to 01/23/2007 08:42 AM. The file size remains 1,075 bytes.

Date	Time	Type	Name	Size
01/18/2007	11:29 AM	<DIR>	.	
01/18/2007	11:29 AM	<DIR>	..	
09/08/2006	04:01 PM		3,645 apexins.sql	
07/18/2006	06:12 PM		6,698 apexvalidate.sql	
09/08/2006	04:01 PM		9,856 apex_egg_config.sql	
09/21/2006	03:51 PM		3,439 apxdbnig.sql	
08/21/2006	03:51 PM		1,537 apxremov.sql	
07/24/2006	03:59 PM		2,877 apxxenig.sql	
09/08/2006	04:08 PM		1,261 apxxepwd.sql	
01/18/2007	11:29 AM	<DIR>	builder	
09/08/2006	04:02 PM		3,972 catapx.sql	
01/18/2007	11:29 AM	<DIR>	core	
09/08/2006	04:06 PM		85,975 coreins.sql	
09/08/2006	04:02 PM		86,872 coreins2.sql	
01/18/2007	11:28 AM	<DIR>	doc	
01/18/2007	11:29 AM	<DIR>	images	
09/08/2006	05:59 PM		1,075 load_trans.sql	
01/23/2007	08:42 AM	<DIR>	patch	
01/18/2007	11:29 AM	<DIR>	utilities	
10/10/2006	07:07 AM		3,985 welcome.html	
			12 File(s) 211,192 bytes	
			8 Dir(s) 22,976,561,152 bytes free	

Database Testing

Why is this type of testing required?

Database testing will play a significant role since it will start with the upgrade script. This testing will be done with the Database Administrative team (DBA).

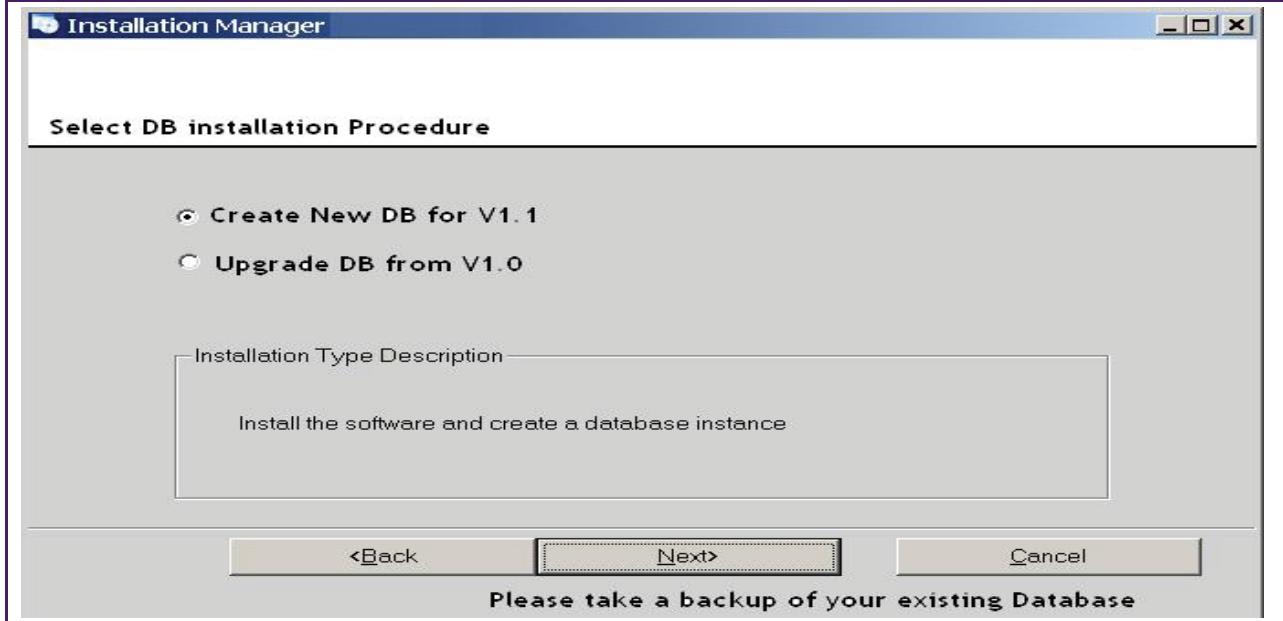
How do we test?

The database instance needs to be stopped or brought down and a backup of the database will be taken and stored. An Upgrade Script will be executed and a new database will be delivered. The tester must open the database log file and check if all the records are inserted or applied correctly. Any database error is reported as a defect.

Most often either a new table or a relation is added to the database; hence the strategy for database testing lies in testing the BLANK database, i.e. the first install the Default Database V1.0. This database will have only the tables and relation and not the "DATA" as such.

Then the tester must apply the alter scripts (Upgrade Script) and check if there are any errors in the log file. If there are no errors, connect the Database with the application and try to execute few test scenarios which have the basic flow of the application. This is done to make sure that the data is correctly being entered into the application and to verify there is no error for inserting or updating any database record.

Exhibit 2: Sample Installation Screen



Application Testing

Why is this type of testing required?

Application testing is performed when both the installation testing and database testing is complete and the application is responding correctly.

How do we test?

Application testing has two separate paths; first is deploy the application V1.0 with a BLANK database, i.e., with the default database, and then upgrade the system to V1.1 and perform the basic flow test.

The second path is to take the application V1.0 which has data in the database, i.e., the Production environment, run the upgrade path and convert the application to V1.1. Check if all data which was present in V1.0 is reflected in V1.1. Also check if the user is able to perform the basic flow actions and the system is performing efficiently.

Application testing also involves testing all the path or counters of the application. For example, suppose the functionality of a module is to upload and store the file at a physical

location. All files which were uploaded earlier need to check for their consistency. Files which are uploaded after the up-gradation need to be verified and the timestamp of the file which is uploaded needs to be checked.

With the Application Under Test (AUT) you will have to modify some of the earlier records. For example, suppose you are testing a web-based banking application which has payments associated with it. During Up-gradation testing, you will need to open a Payment record which was created in V1.0 and modify it. The tester should be able to perform this action since this simple flow might involve data interaction with many Tables and primary and secondary constraints.

Documentation Testing

Why is this type of testing required?

Documentation testing provides verification of the Release Notes or Installation procedures associates with an upgraded application. It is performed after the application is tested in all respects. This plays a significant role since customers often read the Release Notes or Installation procedures to install or upgrade their system.

How do we test?

Testing the documentation involves understanding the detail procedures and any workaround or any known issues which are still open. For example, the upgrade user should be advised to take a backup of the existing database, all the uploaded files or configuration files as applicable based on the upgrade. The release notes may require specific notes and instructions for copying requested/required files to a new location. All pre-requisites which should be written into the release notes or installation instructions should be uncovered here. The dos and don'ts of installation must be verified before the installer can be delivered to a customer.

Summary

Nearly 60 percent of software development projects are considered maintenance and these usually require either a Hot Fix or Patch Version to be delivered to a customer. Knowing that different customers are on different versions or your application means Up-gradation Testing must play an important role to minimize errors. Upgrade Path Testing includes Installation, Database, Application and Documentation testing and can help minimize defects in the installation of new patches and fixes. Such testing should be well defined with checkpoints established ahead of time.

ABOUT THE AUTHOR

Dhiraj P. Lokhande is a Senior Consultant in Kanbay's Testing practice and has almost 8 years experience in testing financial services systems and applications. He can be reached at dzlokhande@kanbay.com.

ABOUT KANBAY

Founded in 1989, Kanbay International, Inc. (NASDAQ: KBAY) is a global IT services firm with over 6,900 associates worldwide. Kanbay provides a highly-integrated suite of management consulting, technology integration and development, and outsourcing solutions through a proven global delivery platform to clients focused on Financial Services and Consumer and Industrial Products, as well as an emerging presence in the Communications and Media and Life Sciences industries. Kanbay is a CMM Level 5 assessed company headquartered in greater Chicago with offices in North America and India as well as London, Singapore, Hong Kong and Melbourne.