Robust Automation Framework using Database for Test Data

Merral Crasto
Testing Competency
IBM India Pvt Ltd
Bangalore, India

Sriharirao Kuchi
Testing Competency
IBM India Pvt Ltd
Bangalore, India

Robust Automation Framework using Database for Test Data is a Robust Framework which will simplify and speed up scripting by making use of reusability. This is hybrid framework where the whole scripting process requires creation of independent scripts, which represents a small section of the application under test. The framework separates the data from the functionality and hence has low maintenance. Highlight of this framework is here the test data is fetched from Database for a particular script and after the script is executed the data is being updated again in to Database and the results are updated to results database. To handle dynamically generated data and to further synchronize the test data across the applications being automated Database is being used in an hybrid automation framework to easily manage, update, retrieve and synchronize data needed for automation scripts.

Index
A. Robust Automation Framework – Framework Functioning
B. Robust Automation Framework – Framework Elements
C. Robust Automation Framework – Steps to Automate Scenario
D. Robust Automation Framework – Benefits

Robust Automation Framework Using Database for Test Data
Software application undergoes rigorous functional tests, especially supported by automated testing frameworks. Automating these frameworks and maintaining quality software releases are critical to business performance. By implementing the appropriate testing framework, enterprise can significantly increase the speed and accuracy of the testing process, provide a higher return on investment from Software project and systematically minimize risk.

The Test Automation Framework must be modular, easy to maintain, reusable, and leveraged across multiple projects to maximize the value of test Automation Suite
Common automation considerations for a Test Automation Framework include:

- Maximizing test coverage
- Scale for future requirements
- Reliability and Consistency of result
- Duration of execution
- Identification of all regression testing defects
- End to end testing capability to cover business scenarios

This abstract gives a brief description about a Robust Automation framework, which helps in designing scripts independent of the application. Highlight of this framework is here the test data is fetched from Database for a particular script and after the script is executed the data is being updated again in to Database and the results are updated to results database. To handle dynamically generated data and to further synchronize the test data across the applications being automated Database is being used in an hybrid automation framework to easily manage, update, retrieve and synchronize data needed for automation scripts.

A. Robust Automation Framework- Framework Functioning

Robust Automation framework is an attempt to simplify and speed up scripting by making use of reusability. It is a tiered organization of the function libraries. The functions are organized into separate libraries, to help in keeping the functions more managed, thus ensuring reusability. These functions are classified based on the modules in the application. To handle dynamically generated data and to further synchronize the test data across the applications being automated Database is being used in an hybrid automation framework to easily manage, update, retrieve and synchronize data needed for automation scripts.
Why Use Database?

- **Centralized Database** used for maintaining input/output Test Data instead of Datasheets/ Data pools: Many Automation projects face difficulties in maintaining multiple set of test data (Dev/Production). And if the data is generated dynamically and if the data is required to execute other scripts it is difficult to execute by storing values in excel sheets.

  Ex: In a telecom application say if the Cell Number is generated dynamically and the generated Cell Number is to be used further in some other test script, use of database makes it quite simple and reliable.

- **Secure- Confidential Test Data** encrypted: In case of Production data to be used and if some confidential information to be used, data can be encrypted.

  - **Custom Reporting**: Results updated to DB and results can be easily retrieved for any of the Previous/Current releases. This is helpful in retrieving daily Status.

B. Robust Automation Framework – Framework Elements

a) Environment Setting Files  
b) Support-library functions  
c) Database Functions  
d) Test Data  
e) QTP Setting Scripts  
f) Object Repository  
g) Recovery Scenario  
h) Application Function Library  
i) Sync Scripts  
j) Test Scripts  
k) Batch Execution  
l) Reporting
a) **Environment Setting Files**

File written in XML format. Depending on the environment either Production or Dev, we need to load the particular XML file. The settings that would go into these files are URL, DSN name, Login details for different environments, etc.

b) **Support Library Functions:**

Scripts /Reusable Actions will call the functions from the library files. These functions are application (AUT) independent. These are used to perform some basic tasks. As the application revolves around performing common tasks like entering values to the fields, comparing values, retrieving values from the fields, clicking on links, images, and buttons, displaying messages, retrieving row and column counts, and sorting columns, the generic functions can be reused extensively.

c) **Database Functions:**

Application independent functions can be used to handle the data requirement and updating

d) **Test Data:**

Test Data is application specific data used to test the application--under--test (AUT). is here the test data is fetched from Database for a particular script and after the script is executed the data is being updated again in to Database and the results are updated to results database. To handle dynamically generated data and to further synchronize the test data across the applications being automated Database is being used in an hybrid automation framework to easily manage, update, retrieve and synchronize data needed for automation scripts.

e) **QTP Setting Scripts:**

This script is to set the settings for Automation Tool, i.e., settings common to entire project. After loading the script it should set all the necessary controls. For ex: Load the required Add-ins, Synchronization time etc…

f) **Object Repository:**

Object Repository is a collection of objects like link, button, tables, fields, applets, frames. While executing a test condition, these objects are recognized by QTP and mapped to the objects in the application rendering the test as success/failure. The Object Repository contains the objects for the GUI based validation testing. Any change in GUI objects is modified at a central location, thereby avoiding rework of scripts.

g) **Recovery Scenario:**

Recovery Scenarios define and guarantee the robustness of the scripts. They are defined and implemented to handle unexpected exceptions where the script execution may come to a halt. Recovery Scenarios help the automation script flow to follow a definite execution flow or path whenever unexpected exceptions arise during the script execution.
Functions that used to handle unexpected message boxes, unknown popup/recurring popup /Security popup

h) **Application Function Library:**

Function Libraries form the basic building blocks of an automation test suite and define the common set of reusable functions. These functions are application (AUT) dependent. Reusable functionalities related to application go to this file.

i) **Sync Scripts:**

This script consists of actions to synchronize the test data with the Application Data.

j) **Test Scripts:**

Test scripts are a critical component in the Test Automation suite. They contain the actual code logic to perform and validate a specific business or functional scenario. This includes the automation scripts for different Business Scenarios / Test Cases.

k) **Batch Execution:**

Module wise plans to execute bunch of test cases / Business Scenarios.

l) **Reporting:**

A Reporting or Logging Mechanism should be configured so that test results generated during the execution phase are easily accessible and understood. Log generation is an important part of execution. It is very important to generate debug information at various points in a test case. This information can help find problem areas quickly and reduce the bug-fixing time. Here the Results are updated to DB and results can be easily retrieved for any of the Previous/Current releases. This is helpful in retrieving daily Status.

C. **Robust Automation Framework- Steps to Automate a Scenario**

- Establish connection with database using database functions (DB Functions). The connection will be established based on which environment (Dev/Prod) the script has to be run. The environment where the script has to be run is specified in Environment file i.e either Dev or Prod.

- If script has to be run on Dev environment then the Dev Environment file (.xml file) will be launched which has all the details of Dev environment which includes Dev Data Username/Password, DSN Name, URL’s of Dev environment and Login credentials.

- Data is picked from database. Data is picked from multiple tables in database (Database Ex: MSSQL/Oracle) using Database functions.

- Based on the required data certain steps are performed in the Automation tool, where application dependent (Application function library) and application independent (Support Library) functions perform certain actions.

- Once all the required steps are performed in the particular script the latest data in the application is then updated back to the database (Database Ex: MSSQL/Oracle).

- Once Script is complete Results are updated in Results Database, which can be used for custom reporting.

- The Database connectivity is disconnected.
D. Robust Automation Framework- Benefits:

a) **Key Features:**
- Robust Automation framework which will speed up and simplify scripting.
- Application Independent Database functions.
- Application Independent Support Library Functions
- High Reusability and Low Maintenance.
- Reusability– The framework with its existing set of functions can be used for automating any set of applications

b) **Key Differentiating Features:**
- Low maintenance – Existing framework can be modified with minimal effort.
- Centralized Database used for maintaining input/output Test Data instead of Datasheets/ Data pools.
- Secure- Confidential Test Data encrypted.
- Custom Reporting: Results updated to DB and results can be easily retrieved for any of the previous releases. Apart from the report generated by the test automation tool the framework supports custom reporting based on the parameters defined during the
requirements stage of automation. This helps the QA analysts to quickly know the behavior of the application after an execution which is easy to read and understand from the application perspective.

ACKNOWLEDGMENT

I am heartily thankful to IBM Testing Competency, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.
I wish to express my sincere gratitude to Mr. Ramesh Sn Vasamsetti and Mr. Srinivas Mangipudi. Without their encouragement and guidance this Paper would not have materialized.
Lastly, I offer my gratitude to all of those who supported me in any respect during the completion of the Paper.

Merral Crasto joined IBM in 2006 and he has 4+ years of experience in Software Testing. He is CSTE and ISTQB certified Test Professional. He is a Sr. Test Engineer in IBM. He has worked in ERP, Telecom, Retail domains. He has experience in both Manual and Automated Testing.

Sriharirao Kuchi joined IBM in 2005 and he has 10 years of experience in IT Industry. He has IBM Certified Accredited IT Specialist and Technical resource in IBM. He is a Test Lead in IBM. He has worked in ERP, Telecom, Health care domains. He has experience in both Manual and Automated Testing.