NEKST IT
TESTING EXCELLENCE COMPANY

WHITEPAPER

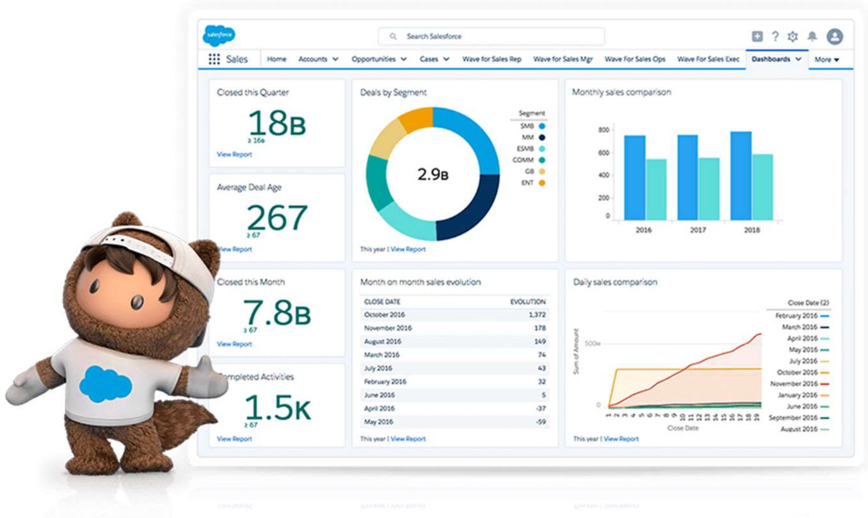
Scalable and Maintainable Salesforce Test Automation with Provar

INTRODUCTION

The Salesforce universe is rapidly maturing. In recent years, the testing of Salesforce implementations has received increasing attention. The growing number of tests and the subsequent professionalization of testing environments have also led to a growing interest in Salesforce test automation. There are various (free) tools for automated testing on the market, each with its own specific pros and cons. Based on years of experience in this field, Nekst IT has named Provar as the undisputed number one solution on its Salesforce test tooling list.

In recent years, Provar and Nekst IT have realized many successful implementations for a wide variety of clients. In this whitepaper, you will find several tips and tricks to kickstart your Provar implementation in a smart and effective manner.

Dimitri Fioole
Chief Operating Officer Salesforce



NEKST IT BV

Westluidensestraat 55
4001 NE Tiel, The Netherlands

+31 (0)85 303 5697
salesforce@nekst-it.com

PREFACE

It's a situation that many of us instantly recognize: you have a toolset that keeps on growing. At a given moment, the software is altered. The impact can be disastrous. Tests suddenly stop functioning properly, whilst analyzing and adjusting your testing data become increasingly difficult tasks. Carefully assessing your test setup, thereby considering the maintainability, expandability, and portability of your tooling, can prevent a lot of trouble.

Taking the following tips to heart will make your life infinitely easier. These simple rules give you more time to develop new test cases and reduce the time invested in maintenance tasks.

1. MAKE GOOD USE OF EXTERNAL DATA

You're well advised to refrain from putting hard-coded data in your test cases. The problem with this mode of operation? It makes the test case less useful because it assumes that information, which should be flexible, is actually fixed and unchanging. This creates issues if you want to extrapolate a certain test case to a new case that uses different data.

We can illustrate this with an example. Say that a certain account contains different statuses, names, and addresses. Chances are that you want to test all these pieces of information separately, without turning them into hard-coded test cases. This could very well mean that you must go through every single test case to adjust a single field name.

Using external data streamlines the operation and allows you to develop and apply one test case that can be used for different scenarios. In my explanation, I will use the data shown below and limit the number of input fields to keep things accessible. When you make real-life test cases, you'll obviously use significantly more input fields.

	A	B	C	D
1	AccountLineNumbe	AccountName	Type	
2	1	Jan	Prospect	
3	2	John	Customer - Direct	
4	3	B&B	Customer - Channel	
5	4	Smith	Channel Partner / Reseller	
6	5	Nekst IT	Installation Partner	
7	6	Claudia	Technology Partner	
8	7	Smelly cat	Other	
9	8	SuperHero	Prospect	
10	11	Discovery	Installation Partner	
11	11	James	Prospect	
12	11	Central perk	Prospect	
13				
14				
15				

Accounts

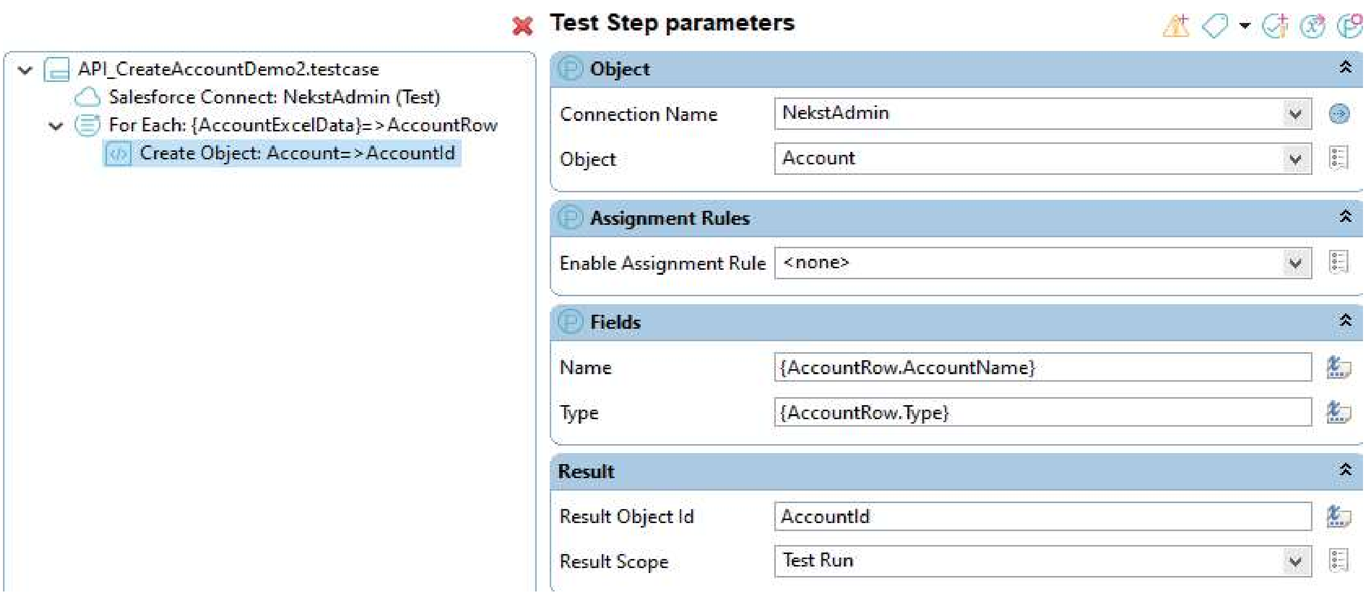
NEKST IT BV

Westluidensestraat 55
4001 NE Tiel, The Netherlands

+31 (0)85 303 5697
salesforce@nekst-it.com

2. USING ALL THE LINES AT ONCE

Do you want to cover multiple scenarios with one single test case? Then you can use the “For Each” functionality in combination with your external data. This functionality then reads a list of data and executes the actions that are listed under the header “For Each” per line of the Excel file that is shown on your screen. This is an effective strategy if you want to double-check the options for creating a wide plethora of different account types.



3. USING ONE SPECIFIC LINE

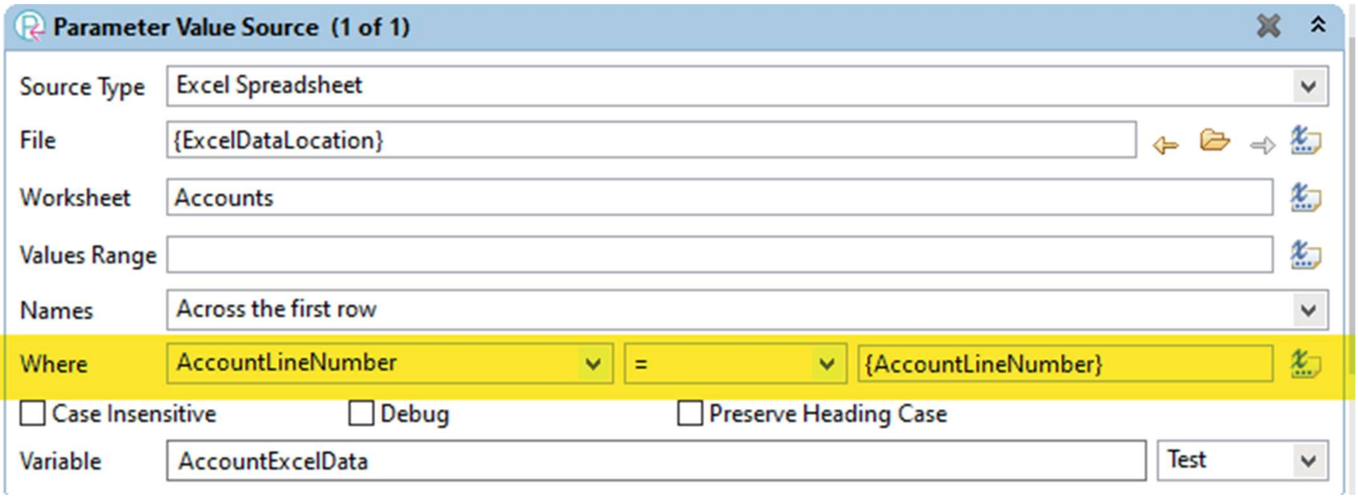
You don’t always want to use a whole set of data for a particular test case. In some cases, a single line will suffice. An end-to-end test, which allows you to link an account to an opportunity, only requires one instead of eleven accounts.

The “Parameter Value Source” allows you to designate which data you want to use by indicating the values that the information should meet. This can be a specific status, name, or number. Using this feature allows you to use the same set of data but filtered according to your specific wishes. This comes in handy if you want to test the mandatory fields only, but also if you want to fill out every single field.

NEKST IT BV

Westluidensestraat 55
4001 NE Tiel, The Netherlands

+31 (0)85 303 5697
salesforce@nekst-it.com



4. USING A SET OF RULES

In addition to focusing on one line or all the lines of your data set, Provar also allows you to use a data set. Using the same condition (the same number or type) enables you to utilize all the rules that meet this specific criterion. In this case, the test equals the "All-At-Once" version. The only thing that has been adjusted is the "Parameter Value Source".

You can use the "IF" statement to ensure that the aforementioned mode of operation is used to the best avail. For example, I use this statement to link products to opportunities. Often, the goal is to link one or several specific products.

5. USING LOCALIZED TEST DATA IN COMBINATION WITH EXTERNAL DATA

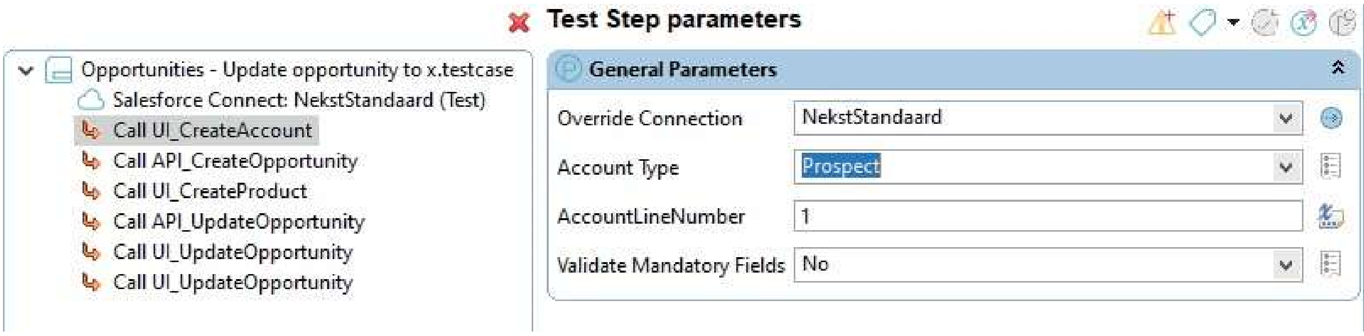
You can also provide data with a variable upfront. This makes sense when you want to use an account from an Excel file that has a different status or type. The main advantage of this method is that your test cases gain even more flexibility, while you need less test data in your Excel file. This makes test case management a lot easier and less time-consuming.

We have entered the status in our example Excel file. You can overrule the displayed status by addressing a variable to the file that determines its new status. This can be the default status mentioned in the existing Excel file, but it can also be a predetermined status or a status that is derived from the Excel data. The possibilities are endless.

NEKST IT BV

Westluidensestraat 55
4001 NE Tiel, The Netherlands

+31 (0)85 303 5697
salesforce@nekst-it.com

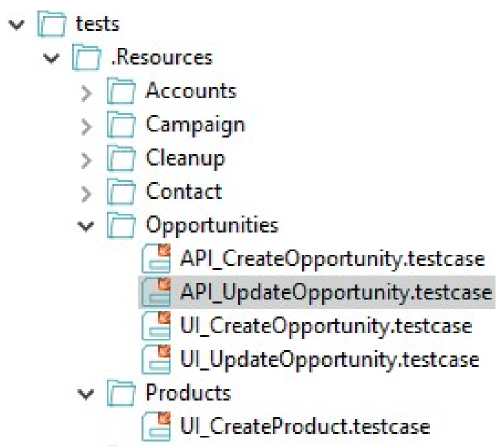


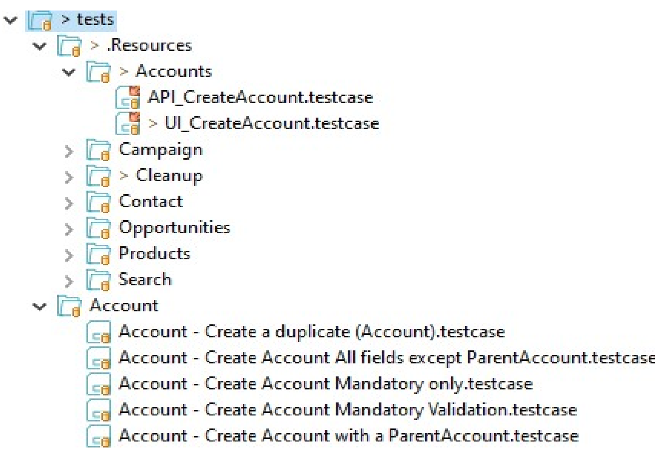
6. MULTIPLE USERS

You can also determine which employee should carry out the test case. Use the "Override Connection" to do this. This functionality allows you to create an end-to-end test case that covers the entire process with different users instead of dropping the entire workload on the plate of the admin. This allows for a far more realistic test case.

7. GENERATING TEST CASES WITH TEST CASES

If you have separated the data from your test cases, it is also very easy to make smaller test cases per functionality or action and link these to your scenario test case. If a functionality is adjusted, you only need to adjust the functionality test case and not the entire scenario test case. In addition, you can easily have multiple functionalities and even the same calling functionality in one test case.

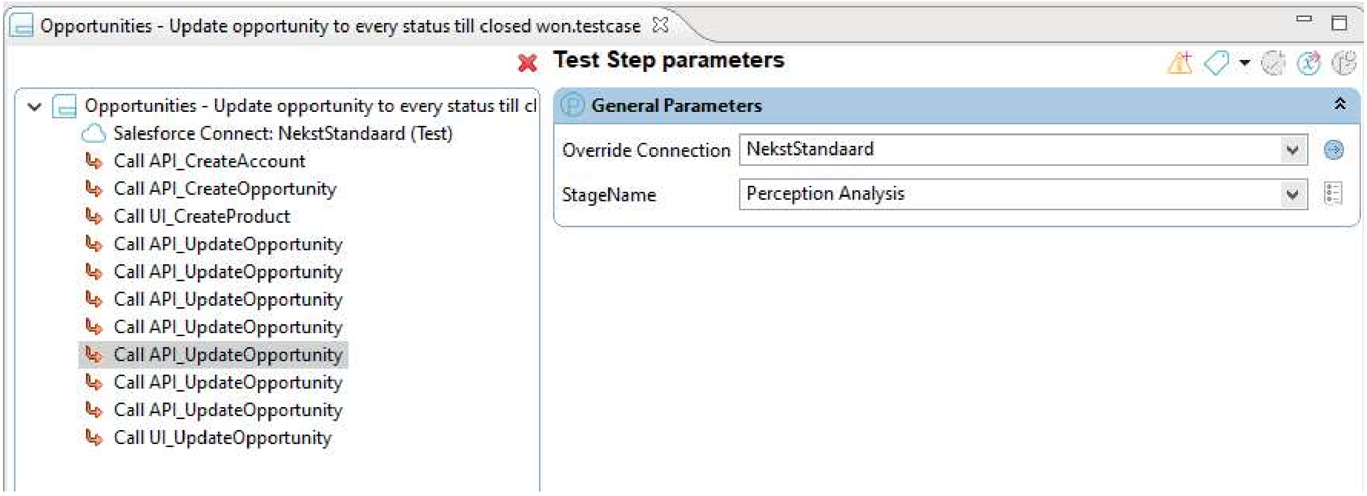




NEKST IT BV

Westluidensestraat 55
4001 NE Tiel, The Netherlands

+31 (0)85 303 5697
salesforce@nekst-it.com



8. MAINTAINABLE TEST CASES

If your test suite gets bigger, you will have to spend more time and resources on maintaining your data bulks. However, carefully assessing the set-up of your test suite(s) can save you a considerable amount of time and money and limits maintenance activities to a bare minimum.

Personally, I like to work with small and modular test cases. Additionally, I keep the scenario test cases and functionality/action test cases separate. This routine allows you to work with smaller reusable modules that you can readily use in multiple places. This way, test cases also remain more maintainable. You do not have to go through all the test cases every time but are able to solely focus on specific functionality test cases. Hence, it is wise to store your scenario test cases and functionality/action test cases separately.

9. KEEP IT READABLE

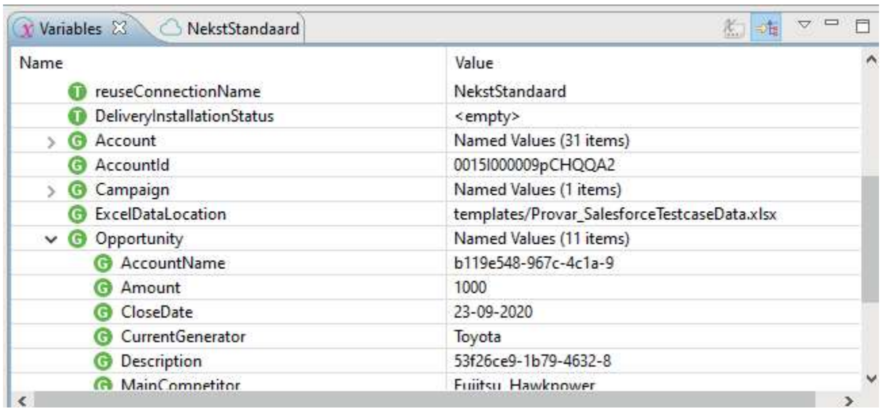
In addition to structure, readability is another important maintenance item. Understanding and (if necessary) adjusting your test cases becomes very difficult if you don't know what you are doing. Therefore, test cases should be given an easily recognizable name. But you should also clearly define the test steps. Provar allows you to effortlessly adjust the screen name and steps. And it gets better! You can also use group steps to create an even clearer and more comprehensible overview. For example, you can group input fields, such as set account information, set address information, and set additional information.

NEKST IT BV

Westluidensestraat 55
4001 NE Tiel, The Netherlands

+31 (0)85 303 5697
salesforce@nekst-it.com

Variables often carry simple names, like AccountName, AccountAddress, Birthday, etc. When you are running the test, the system lists the different variables under "Account", which is a functionality that simplifies and streamlines the searching process.



ABOUT NEKST IT

Nekst IT is a Provar partner and the first Salesforce partner in the world that exclusively focuses on testing and quality assurance. These aspects of IT demand a different approach than developing, configuring, and implementing. Impartially safeguarding the quality of customer systems is our core business.

Nekst IT professionals are positive and proactive people. They are well-trained experts who dare to ask questions, like to investigate exciting new opportunities and aren't afraid to challenge the established order. This mindset allows us to go that extra mile for our customers.

Nekst IT testing professionals excel when it comes to versatility and the intricacies of the testing craft. Based on their broad expertise, their talents in the field of IT testing are deployed in challenging projects for a wide variety of clients.

Do you want to know more about Salesforce testing? Then don't hesitate to contact us. Send an email to **salesforce@nekst-it.com** or give us a call on **+31 (0)85 303 5697**.

NEKST IT BV

Westluidensestraat 55
4001 NE Tiel, The Netherlands

+31 (0)85 303 5697
salesforce@nekst-it.com