

## SAP testing

### 1 Introduction

Generally software testing consists of the disciplines Test analysis, Test automation and Test management. The next sections present these disciplines in respect to SAP. The Autom8-IT test suite can be used during different test levels of a SAP® implementation. These test levels are unit test, integration test, system test, acceptance test and regression test. The next section takes several steps of a Sales and Distribution scenario as an example to get the reader familiar with keywords, the structure, the format and layout of Autom8-IT test scripts. [This short introduction](#) describes an easy, non SAP, example of keywords. The beginning of the next section reviews the easy SAP keywords example in [User instruction guides](#). Hence we complete the section from a software testing perspective with a more complicated keyword, corresponding to the SAP transaction: create sales order.

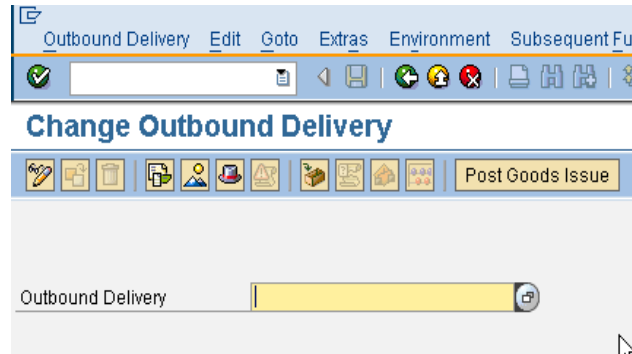
### 2 Test analysis SAP

Keep in mind that the presentation of a SAP test case is always a struggle between the combinations readability and missing functionality and less readability and unambiguous functionality! Autom8-IT prefers the latter, because it prevents hesitation, uncertainty whether fields need to be populated or not, which (operation) keys need to be pressed and a re-test can be reproduced similarly (in case of defects). Also a big advantage is that in case of a failed test script it is indisputably true that either the test script contains a fault or (the software at) the test environment has a defect. Let's start explaining the keyword principle for SAP! The table below shows the last part of a sales and distribution script. The first part of the sales and distribution script consists of two test cases, "create sales order" and subsequently "create and pick a delivery". They are both omitted here. The table lists the remaining procedure of the "Order to Cash"-scenario a user must follow. Starting at the upper left corner and reading till the end of the line gives the following clarification per row:

<b>test case</b>	<b>PGI</b>							
<b>import variable</b>	VL01N							
	<b>Session handle</b>	<b>Outbound delivery</b>	<b>Operation key</b>					
<b>vl02n</b>	&myHandle	&VL01N	Post Goods Issue					
<b>test case</b>	<b>Create billing document</b>							
	<b>Session handle</b>	<b>Billing type</b>	<b>Serv.rendered date</b>	<b>Billing date</b>	<b>Pricing date</b>	<b>SD document</b>	<b>Operation key</b>	<b>Operation key</b>
<b>vf01</b>	&myHandle	&Empty	N/A	&Date("DDMMYYYY")	N/A	&VL01N	Enter	Save

- i) A standard keyword "test case". The "test case" description parameter has the value "PGI." This is the description of the SAP process that is covered in the test case: Post Goods Issue for a delivery
- ii) A standard keyword "import variable". The "import variable" name parameter is "VL01N." This is the a priori unknown delivery number generated by the SAP-system in the test case "create and pick a delivery". Its value is used in the next keywords "vl02n" and "vf01"
- iii) A comment line at the third row. A comment line has a blank cell in the first column, the keyword column. In this case the comment line describes the three parameters of the next keyword "vl02n": "Session handle", "Outbound Delivery" and "Operation key" in the second, third and fourth column respectively

- iv) The keyword “vI02n” specified by a test analyst. The keyword is called “vI02n”, because it corresponds to the (technical) SAP transaction name: “Change outbound delivery”. From now on all SAP-related keywords correspond to the technical SAP transaction name. The user must start up transaction “vI02n” in SAP. The variable parameter session handle is a “pro forma” parameter in a manual test but in an automated test it represents the name of the SAP system, the SAP client number, user name and his password. This variable is very useful in security testing with an automated tool or the testing of Workflow. The second parameter is the variable outbound delivery number, imported under *ij*). The figure shows that the “Outbound Delivery” field is the only field at the first screen when starting transaction “vI02n”. After entering the delivery number in SAP the user must press the key/button “Post Goods Issue”, the third parameter of the keyword
- v) A standard keyword “test case”. The “test case” description parameter has the value “Create billing document.” This is the description of the subsequent SAP process that is covered in the next keyword
- vi) Again a comment line describing the parameters of the next keyword “vf01”. The number of parameters of “vf01” is larger than the number of parameters of “vI02n”, because a user can enter more fields in the transaction “vf01”
- vii) The keyword “vf01” specified by a test analyst. The unambiguous instruction/procedure becomes evident by the values of some parameters in this keyword. The values &Empty, N/A and &Date(“DDMMYYYY”,14,0,0) tell the user explicitly what to do! The first value tells the user that he must empty an already populated field or leave it empty. The second one, N/A, tells the user to skip/ignore the field (and keep the value). The third value prescribes an eventual date calculation, which must be entered in SAP in accordance with the prescribed data format. After the field manipulations the user creates the invoice by following the operation key parameter instructions



According to the test script the software tester must execute two test cases during test execution. He must perform a post goods issue and subsequently create a billing document in order to complete the sales and distribution process. Eventually a SAP test analyst might enhance the test script with verification steps of material movement and of the created output of the invoice. Now we arrive at, more serious stuff regarding the number of SAP fields. After the invoice the test analyst could have thought of incidents during shipping with damage consequences of the ordered products. The software tester must create a sales order of the sales document type “CR” - a credit memo request - with reference to the created billing document. The table below shows an example with the following keywords:

- i) A standard keyword “test case”. The “test case” description parameter has the value “Credit note with reference to billing document.” The keyword “va01” covers this SAP business process
- ii) A standard keyword “import variable”. The “import variable” name parameter is “VF01” This is the a priori unknown billing document number generated by the SAP-system in a previous test case. Its value is used in the next keyword “va01”
- iii) A comment line describing the parameters of the next keyword “va01”
- iv) A test analyst defined keyword “va01” with a large number of SAP-parameters spread over five rows; Therefore comment line and keyword row, still belonging to the “va01”-keyword, change places till the end of the table. The large number of parameters - each parameter corresponding to a SAP field description within transaction “va01” - implies an unambiguous notation. The unambiguous notation prevents hesitation, uncertainty whether fields need to be populated or not, which (operation) keys need to be pressed and reduces the chance on deviations from the business process during test execution

test case	Create credit note with reference			to billing doc				
<b>import variable</b>	VF01							
	<b>Session handle</b>	<b>Sales document type</b>	<b>Sales organization</b>	<b>Distribution channel</b>	<b>Division</b>	<b>Sales office</b>	<b>Sales group</b>	<b>Operation key</b>
<b>va01</b>	&myHandle	CR	&VKORG	10	00	1000	101	Create with Reference
	<b>Sales document</b>	<b>Req dd type</b>	<b>Req deliv.date</b>	<b>Operation key</b>	<b>Sold-to party</b>	<b>Ship-to party</b>	<b>PO number</b>	<b>Req dd type</b>
	&VF01	N/A	N/A	Copy	N/A	N/A	&Scen	N/A
	<b>Req deliv.date</b>	<b>Delivering plant</b>	<b>Complete delivery</b>	<b>Delivery block</b>	<b>Billing block</b>	<b>Pricing date</b>	<b>Payment card type</b>	<b>Card number</b>
	N/A	N/A	N/A	N/A	&Empty	N/A	N/A	N/A
	<b>Expiration date</b>	<b>Payment terms</b>	<b>Incoterms</b>	<b>Incoterms (part 2)</b>	<b>Order reason</b>	<b>Material</b>	<b>Order quantity</b>	<b>Sales unit</b>
	N/A	N/A	N/A	N/A	200	N/A	N/A	N/A
	<b>Item category</b>	<b>Plant</b>	<b>Reason for rejection</b>	<b>Division</b>	<b>Storage location</b>	<b>Operation key</b>		
	N/A	N/A	N/A	N/A	N/A	Save		

The second parameter “sales document type” and the first “operation key” parameter with the value “Create with Reference” determine the logic, which the software tester within the transaction must follow. Right after the first operation key the value of the sales document number “&VF01” needs to be entered in SAP, i.e. the just created invoice in the SAP-system. As a result of pressing “Copy” for the second operation key, the data of the reference billing document in SAP is copied into the new order. All parameters except the order reason can have the value N/A. The keyword “va01” illustrates that software testers might need a little push forward, given by an experienced SAP test analyst, in the overwhelming number of fields to choose from in SAP. The table below and the next paragraph discuss the same keyword for a standard order type.

test case	Create sales order							
	<b>Session handle</b>	<b>Sales document type</b>	<b>Sales organization</b>	<b>Distribution channel</b>	<b>Division</b>	<b>Sales office</b>	<b>Sales group</b>	<b>Operation key</b>
<b>va01</b>	&myHandle	OR	&VKORG	10	00	1000	101	Enter
	<b>Sales document</b>	<b>Req dd type</b>	<b>Req deliv.date</b>	<b>Operation key</b>	<b>Sold-to party</b>	<b>Ship-to party</b>	<b>PO number</b>	<b>Req dd type</b>
	N/A	N/A	N/A	N/A	&KUNNR	&KUNNR	&Scen	D
	<b>Req deliv.date</b>	<b>Delivering plant</b>	<b>Complete delivery</b>	<b>Delivery block</b>	<b>Billing block</b>	<b>Pricing date</b>	<b>Payment card type</b>	<b>Card number</b>
	&Date(")	&Plant	N/A	N/A	N/A	N/A	N/A	N/A
	<b>Expiration date</b>	<b>Payment terms</b>	<b>Incoterms</b>	<b>Incoterms (part 2)</b>	<b>Order reason</b>	<b>Material</b>	<b>Order quantity</b>	<b>Sales unit</b>
	N/A	N/A	N/A	N/A	N/A	&MATNR	1	N/A
	<b>Item category</b>	<b>Plant</b>	<b>Reason for rejection</b>	<b>Division</b>	<b>Storage location</b>	<b>Operation key</b>		
	N/A	N/A	N/A	N/A	N/A	Save		

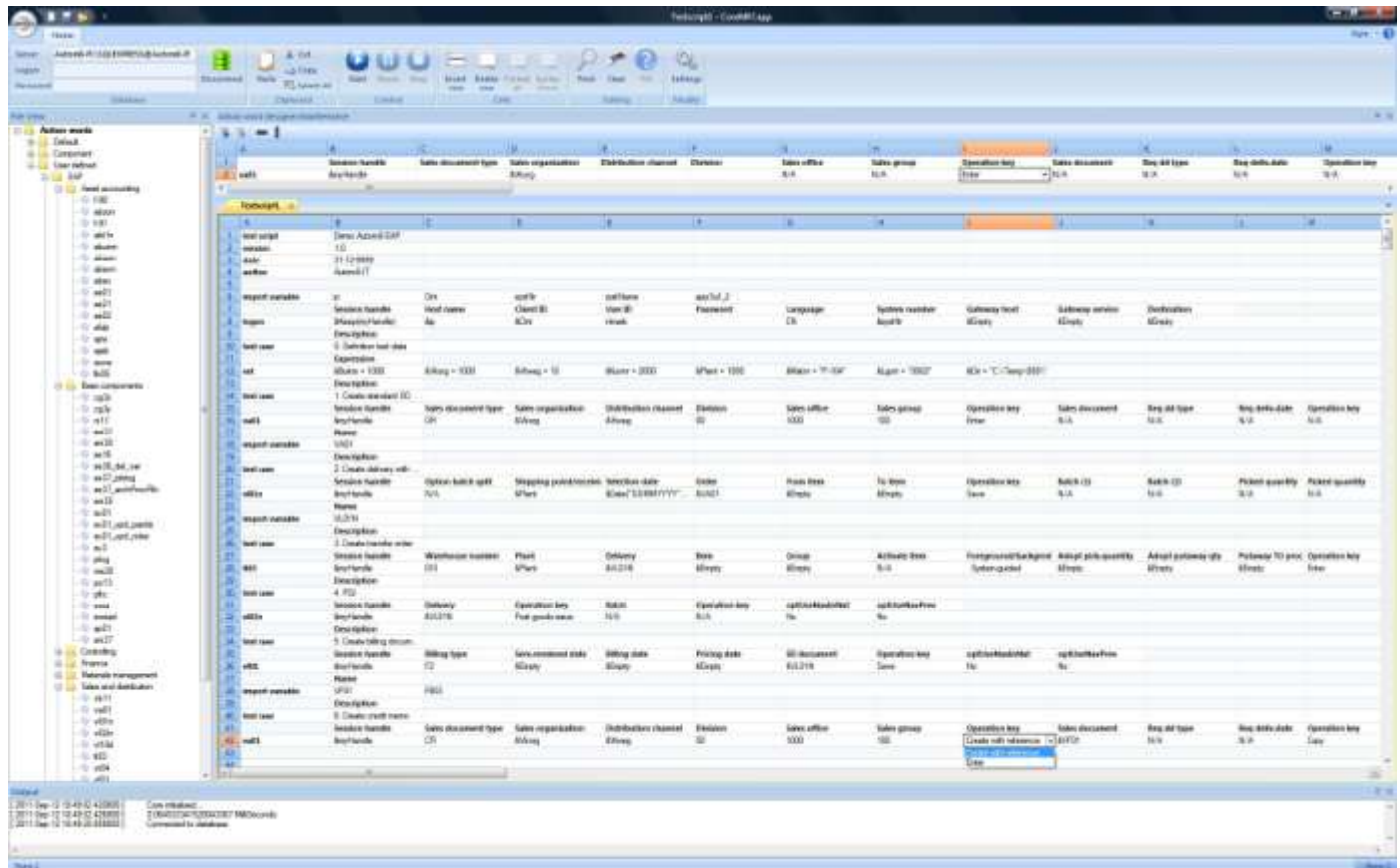
Apparently, the “CR” order type is not the only sales order type. The standard sales and distribution process begins creating a standard sales order with order type “OR”. Keeping in mind the large number of SAP transactions then it is not very wise and pleasant from maintainability and configuration management perspective to start creating a new keyword for every different sales order type. As every little change, functional or test related, for the sales order needs to be maintained in all sales order related keywords and corresponding testing programs and test scenarios. Therefore, very consistently, the same keyword is used again when the software tester must create a standard sales order. The tables at the previous page illustrates this.

It appears that the first and second operation key function both as a variable in the keyword “va01”. The value “Enter” of the first operation key for the standard order type determines implicitly that the next four parameters are redundant and have the value N/A, including the second operation key parameter. The combination of both operation key values implies for the corresponding transaction program, used in an automated test with this tool, that the screen with the “reference to a sales document” becomes redundant. Filter functionality in the Autom8 SAP test automation removes it from the actual testing program.

### 3 Autom8-SAP

#### 3.1 Semi-automation SAP test analysis

Using Autom8-SAP in combination with our central test application Autom8 and Improve Testing results in improvement and acceleration of the writing of SAP test scripts. This section describes the automation of writing the SD-test script in the previous section with Autom8-IT and Autom8-SAP. The screenshot below of Autom8-IT shows that the test application window consists of three panes and a menu bar.

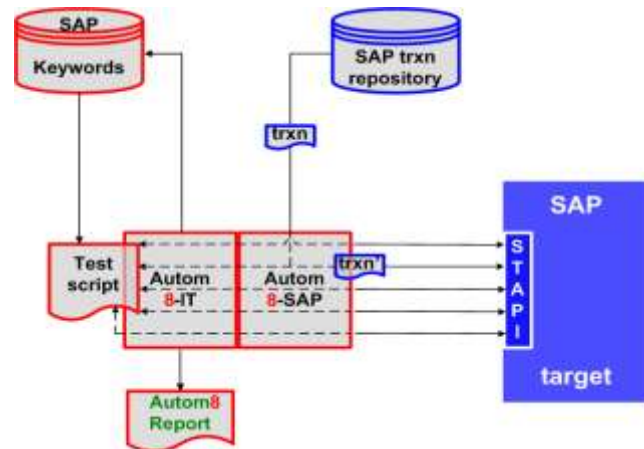


The left-hand pane lists a tree of keyword categories and keywords. The three highest categories are Default, Component and User defined. SAP and the SAP-modules, of which AA, BC and SD are

expanded, are sub-categories of the User defined-category. By "dragging & dropping" keywords from e.g. the SD keyword category onto the test script editor - the main multi-line grid - test scripts can be easily generated. The dynamic "drag & drop" does not come up to the mark due to the static character of this document. In a split second the keyword is dropped onto the test script with pre-defined default test values, so that only relevant test values needs to be chosen from combo box cells and/or eventually empty cells need to be populated with test data / variables. When keywords do not yet exist or miss a (default) parameter(value) the "Keyword designer/maintenance" pane just below the menu bar gives the tester the opportunity to manipulate the content of the (portable) keywords in the keyword category. Consequently the new created or freshly updated keyword can be dragged and dropped again on the test script editor.

### 3.2 Automated SAP testing

The functional test set of an integration test or system test might become quite large as it tries to cover the main relevant scenarios with high business risks. Re-testing of all tests does occur and might be necessary because of last-minute implemented changes or implementation of the SAP software in another test client. However re-testing is not often scheduled in the project planning. Experience learns that automation of tests of non "commercial off the shelf" software pays off after testing the complete test set of a software release seven times. Our estimate for SAP is that automated testing with Autom8-SAP pays off after two or three times, because 90% of the transactions or even more of a SAP-implementation is standard SAP and the testing programs of the SAP transaction repository are portable. On top of that less resources than normally present in a test team can make an immediate start with both writing test scripts - including collecting test data - and test execution after the transport cycle of changes. The adjacent figure shows the architecture of Autom8-IT & Autom8-SAP. As a pair, they are connected to a SAP-system under test. The circle of elements at the left hand side, Autom8-IT, Keyword repository and Test script symbolise the realisation of a test script. Autom8-IT can run the keyword SAP test scripts automatically when the STAPI, Software Testing Application Programming Interface, is implemented at the SAP-system. Analysed top down SAP test scripts are composed of the following low level actions, corresponding to the functionality of our STAPI:



Sql-functionality to retrieve SAP table information

Transaction functionality to process SAP transactions

Authorisation functionality to prove that the SAP-system is secure and audit-proof

Program functionality to run SAP reports and SAP jobs

Input/Output, Interfaces and IDocs functionality to test the tailor-made software