Using mock data in ABAP Unit Test with SAP Standard tools

or

“How do I unit test SAP standard –monster- code which posts data”

ADventas
Consulting

Timo John

- 1980 – Dipl. Wirtschaftsinfromatiker
- Since 2005 SAP development
- Inhouse & Consultant
- ABAP OO
- SD / MM / GTM

Twitter: @Timo_John

Timo.John@ADventas.de

https://www.linkedin.com/in/timo-john-1852336a/

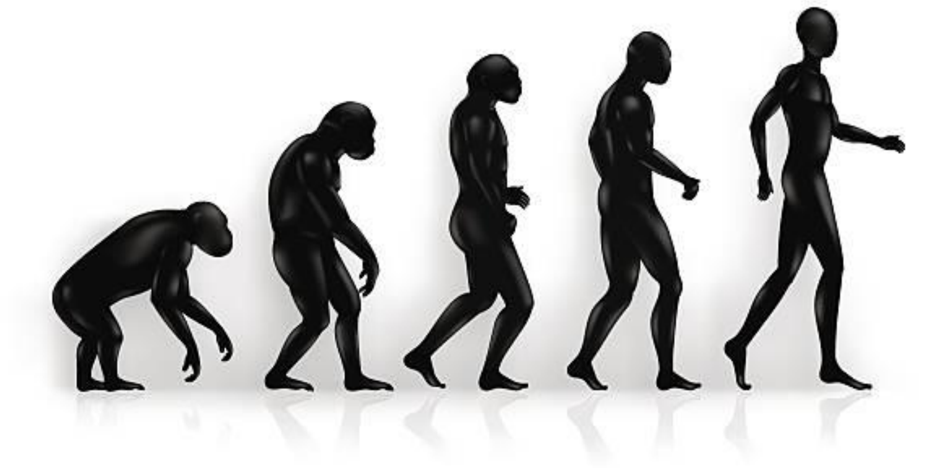
ADventas

ADventas Consulting was founded by Peter Langner in 2006.

As a team we advise and support businesses in implementing application services. ADventas customers are located in the metropolitan region Hamburg as well throughout Europe.

Focus area is international trade in the SAP S/4HANA module GTM

Evolution of unit test coders



1. Simple, local tests
2. Value statements to construct test data
3. “INSERT DB” with test data in unit test
4. Mock everything in Objects approach
5. Real mocked Data

Unit test with dependencies

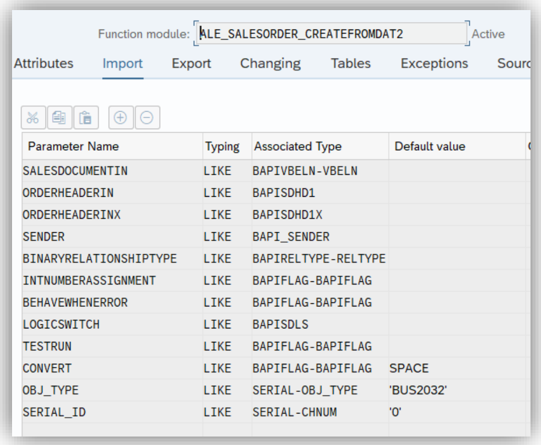
Usage of SAP standard functions is recommended? Sure!

- Simply ignore the dependency and use it
- Create MOCK classes
- Use ABAP Test Double framework
- Use ABAP test-seams
- Use INSERT / UPDATE DB in unit tests prepare phase

But what, if you are dependent on SAP code?

You want to build tests that covers lager parts of a whole process. You are heading towards functional or even end to end tests.

- “I want to test if my enhancement is executed.“
- “I need to post two dependent documents in one test.”
- “I am dependent on SAP code to be executed like bapi_salesorder_createfromdat2 in my process test”.



Function module: ALE_SALESORDER_CREATEFROMDAT2 Active

Parameter Name	Typing	Associated Type	Default value
SALESDOCUMENTIN	LIKE	BAPIVBELN-VBELN	
ORDERHEADERIN	LIKE	BAPISDHD1	
ORDERHEADERINX	LIKE	BAPISDHD1X	
SENDER	LIKE	BAPI_SENDER	
BINARYRELATIONSHIPTYPE	LIKE	BAPIRELTYPE-RELTYPE	
INTNUMBERASSIGNMENT	LIKE	BAPIFLAG-BAPIFLAG	
BEHAVEWHENERROR	LIKE	BAPIFLAG-BAPIFLAG	
LOGICSWITCH	LIKE	BAPISDLS	
TESTRUN	LIKE	BAPIFLAG-BAPIFLAG	
CONVERT	LIKE	BAPIFLAG-BAPIFLAG	SPACE
OBJ_TYPE	LIKE	SERIAL-OBJ_TYPE	'BUS2032'
SERIAL_ID	LIKE	SERIAL-CHNUM	'0'

SAP Standard code is a grey box

- SAP has no exchangeable persistence layer below BAPIs etc.

- You need consistent repeatable test data on DB for every test run.
  - Document numbers should be consistent.
  - Material / Customer settings, Stock and so on…

- There is no simple clever way to set and reset the sap database for unit test from my point of view.

How to –really– test SAP-standard code?

ABAP SQL Test Double Framework & ABAP CDS Test Double Framework

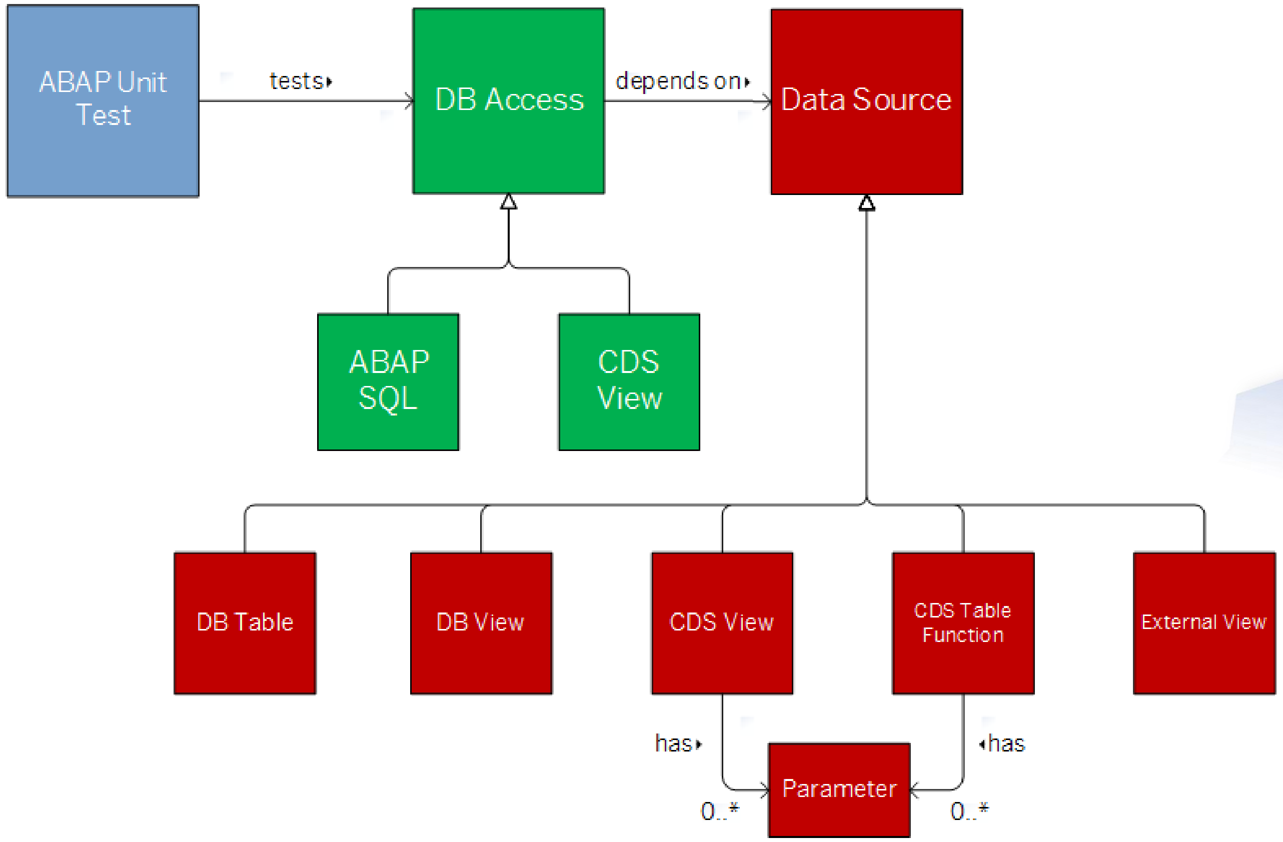
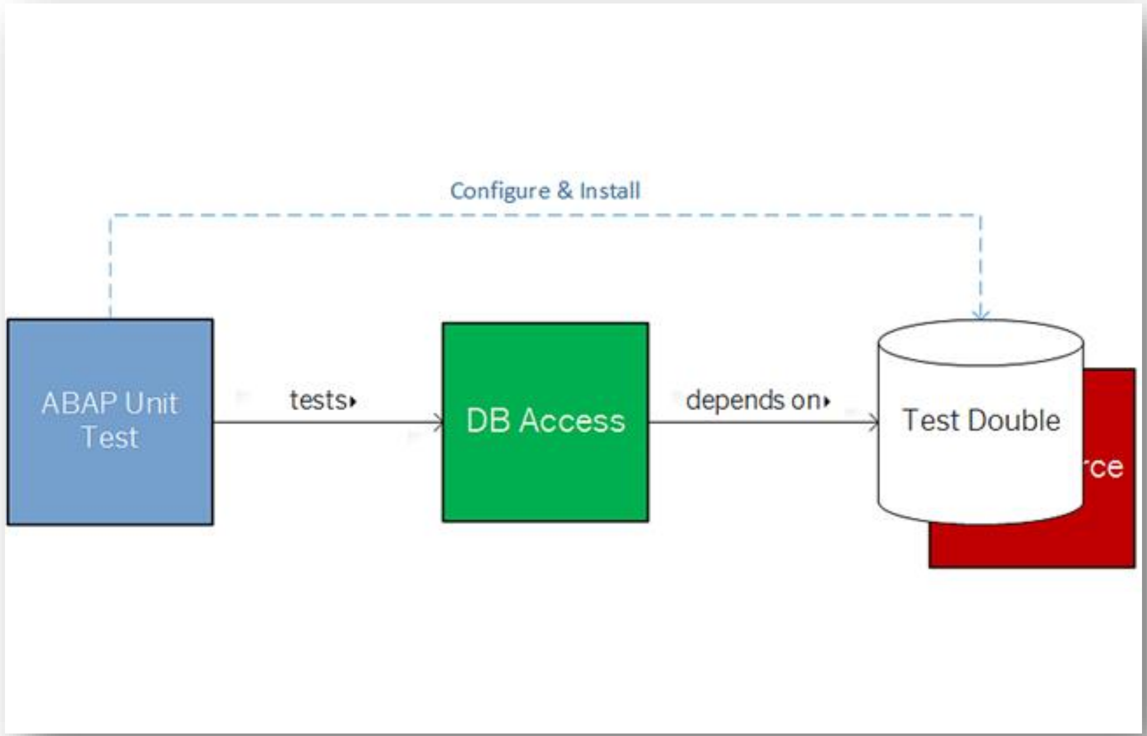


Image source: https://help.sap.com/docs/ABAP_PLATFORM/c238d694b825421f940829321ffa326a/8562b437073d4b9c93078c45f7a64f21.html?locale=en-US

ABAP SQL Test Double Framework & ABAP CDS Test Double Framework

"The frameworks support you in replacing data sources with test doubles, so that the object under test accesses the test double instead of the real depended-on component."



source: https://help.sap.com/docs/ABAP_PLATFORM/c238d694b825421f940829321ffa326a/8562b437073d4b9c93078c45f7a64f21.html?locale=en-US

Why is TDF not sufficient

Using the SAP database double frame work is a great option to simply produce a database constellation of your needs.

What it does not solve is the question where is the required test data stored

A doubled database table is empty!

Live Demo

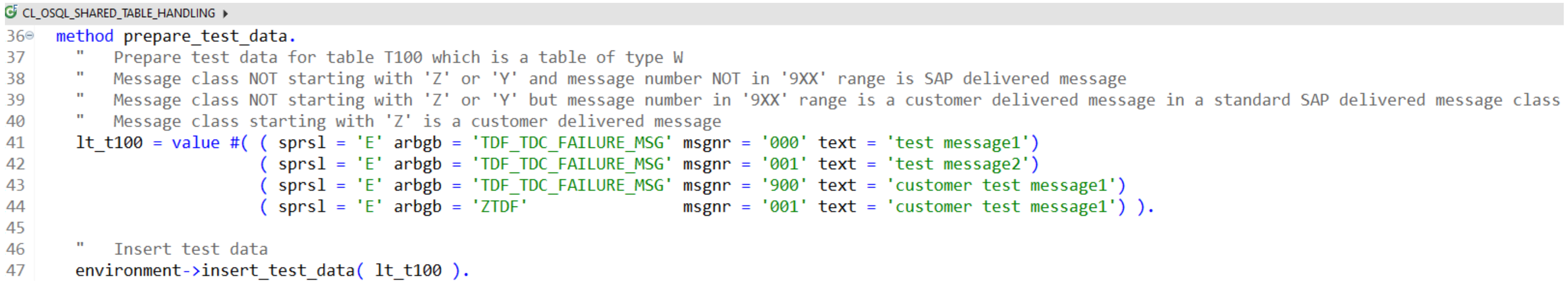
General challenges on unit test data

- Keep them in sync with the unit test
- Avoid accidental changes
- Keep them independent of the underlying system / database

ADventas Consulting

How to handle storage of test relevant data?

Value statements



CL_OSQL_SHARED_TABLE_HANDLING

```
method prepare_test_data.
  "   Prepare test data for table T100 which is a table of type W
  "   Message class NOT starting with 'Z' or 'Y' and message number NOT in '9XX' range is SAP delivered message
  "   Message class NOT starting with 'Z' or 'Y' but message number in '9XX' range is a customer delivered message in a standard SAP delivered message class
  "   Message class starting with 'Z' is a customer delivered message
  lt_t100 = value #( ( sprsl = 'E' arbgb = 'TDF_TDC_FAILURE_MSG' msgnr = '000' text = 'test message1')
                     ( sprsl = 'E' arbgb = 'TDF_TDC_FAILURE_MSG' msgnr = '001' text = 'test message2')
                     ( sprsl = 'E' arbgb = 'TDF_TDC_FAILURE_MSG' msgnr = '900' text = 'customer test message1')
                     ( sprsl = 'E' arbgb = 'ZTDF'                msgnr = '001' text = 'customer test message1') ).

  "   Insert test data
  environment->insert_test_data( lt_t100 ).
```

Pro	Cons
Fast for small amount of values	Very much text in code for complex tests (sap standard IS complex)
Definitions right in the code visible	

ADventas Consulting

How to handle storage of test relevant data?

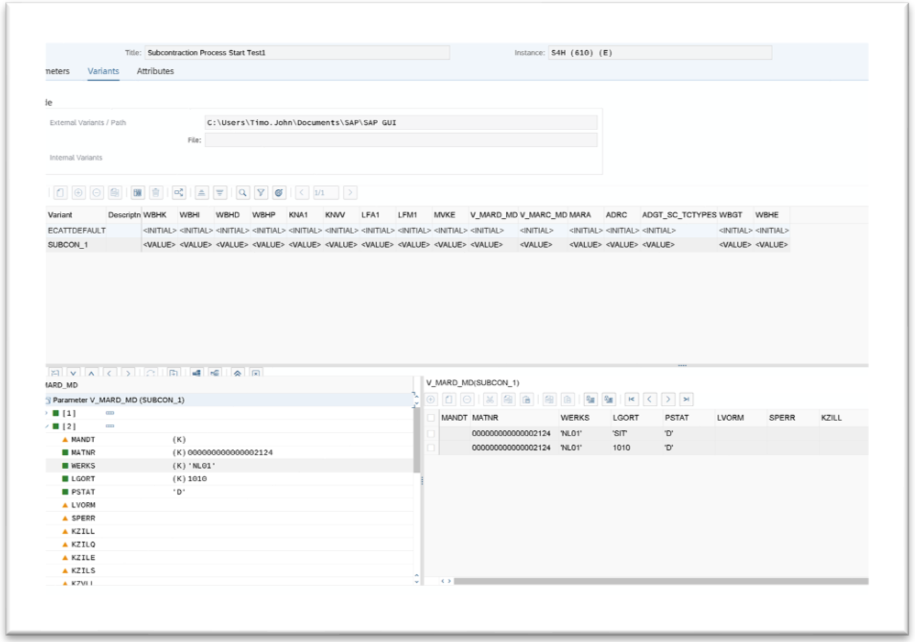
Select and insert data into

```
Select * from T100 using client XXX …
Insert into T100….
```

Pro	Cons
Simple	Usable only in designated scenarios
Self refreshing test data, if helpful	

How to handle storage of test relevant data?

Using ECATT Test Data Container (TDC)



Title: Subcontraction Process Start Test1

Instance: S4H (610) (E)

meters | Variants | Attributes

le

External Variants / Path: C:\Users\Timo.John\Documents\SAP\SAP GUI

File:

Internal Variants

Variant	Descriptn	WBHK	WBHI	WBHD	WBHP	KNA1	KNVV	LFA1	LFM1	MVKE	V_MARD_MD	V_MARC_MD	MARA	ADRC	ADGT_SC_TCTYPES	WBGT	WBHE
ECATTDEFAULT		<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>
SUBCON_1		<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>

ARD_MD

Parameter V_MARD_MD (SUBCON_1)

```
[1]
[2]
  MANDT    (K)
  MATNR    (K) 000000000000002124
  WERKS    (K) 'NL01'
  LGORT    (K) 1010
  PSTAT    'D'
  LVORM
  SPERR
  KZILL
  KZILQ
  KZILE
  KZILS
  KZVLL
```

V_MARD_MD(SUBCON_1)

MANDT	MATNR	WERKS	LGORT	PSTAT	LVORM	SPERR	KZILL
	000000000000002124	'NL01'	'SIT'	'D'			
	000000000000002124	'NL01'	1010	'D'			

Pro	Cons
“Large” data volumes in multiple structures simple to handle.	Initial setup is complex to understand
Can be handled buy a business consultant	No visibility of test data in code

ADventas Consulting

Warning!

Hassle and frustrations is on the road, but YOU keep going!

Database inconstancies might occur!

If the mocked unit test is running for the first time all the pain & frustration is gone!

Puzzle pieces

ABAP UNIT Tests

Report for data collection

SQL Test Double Framework

TDC

Procedure overview 1/2

- first time use regular DB related test help you a lot.
- Collect Tables to mock
  - Tables where test data (master & transactionall) is needed
  - Tables used for written access
- Create test data container in TA: SECATT
  - Build the structure with all needed tables
  - Enable API access
- Fill test data container
  - Customer, material etc you for posting sales order as example

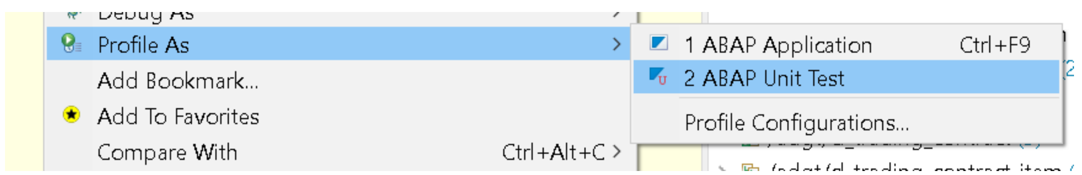
Procedure overview 2/2

- Code Unit test(s) with mock
  - Create DB mocks (in class-setup)
  - Connect TDC parameters with dependencies
  - load container data into mock database.

  - Hint: don‘t forget the number range tables to have consistent document numbers.

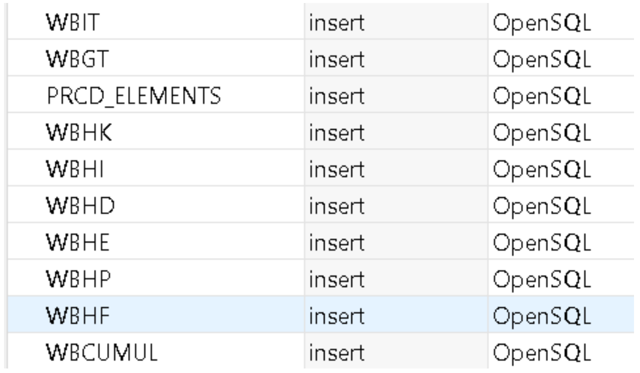
Collection of relevant table

- Use eclipse traces to collect accesses Tables
  - Profile as ABAP Unit Test

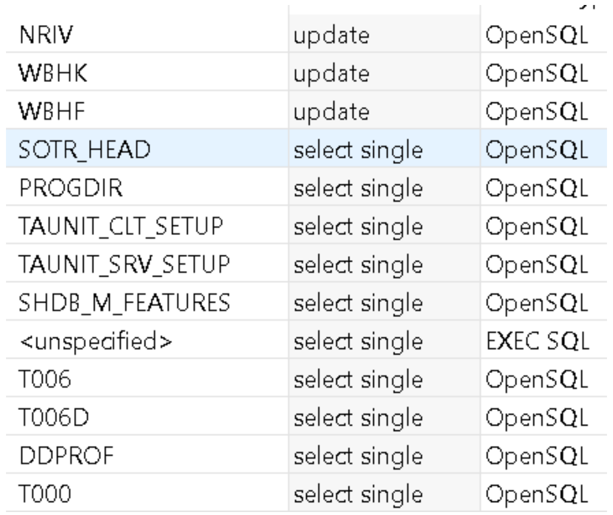


  - Use database access tab

- Group tables:
  1. Mock relevant with test data
  2. Mock relevant write only



WBIT	insert	OpenSQL
WBGT	insert	OpenSQL
PRCD_ELEMENTS	insert	OpenSQL
WBHK	insert	OpenSQL
WBHI	insert	OpenSQL
WBHD	insert	OpenSQL
WBHE	insert	OpenSQL
WBHP	insert	OpenSQL
WBHF	insert	OpenSQL
WBCUMUL	insert	OpenSQL



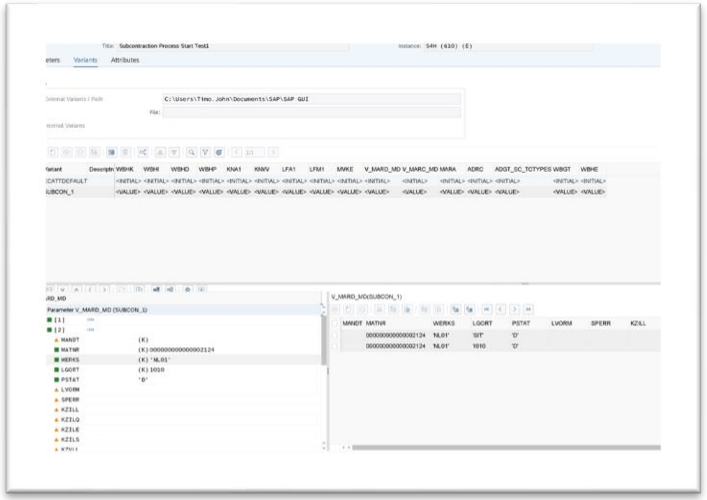
NRIV	update	OpenSQL
WBHK	update	OpenSQL
WBHF	update	OpenSQL
SOTR_HEAD	select single	OpenSQL
PROGDIR	select single	OpenSQL
TAUNIT_CLT_SETUP	select single	OpenSQL
TAUNIT_SRV_SETUP	select single	OpenSQL
SHDB_M_FEATURES	select single	OpenSQL
<unspecified>	select single	EXEC SQL
T006	select single	OpenSQL
T006D	select single	OpenSQL
DDPROF	select single	OpenSQL
T000	select single	OpenSQL

ADventas Consulting

ECATT - Test data container (TDC)

- stores values in DDIC Table types and structures
- Editable via SAPGUI
- Usable on business consultant level to edit.
  - BC duplicates container; edits values for using other material / stock / etc

- Variants in the same TDC can be used to store data for different scenarios (materials / stock situation …)
- Up- / download, transport, RFC …



Title: Subcontraction Process Start Test1
Instance: S4H (610) (E)

External Variants / Path: C:\Users\Timo.John\Documents\SAP\SAP GUI
File:
Internal Variants

Variant	Descriptn	WBHK	WBHI	WBHD	WBHP	KNA1	KNVV	LFA1	LFM1	MVKE	V_MARD_MD	V_MARC_MD	MARA	ADRC	ADGT_SC_TCTYPES	WBGT	WBHE
ECATTDEFAULT		<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>
SUBCON_1		<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>

RD_MD
Parameter V_MARD_MD (SUBCON_1)
[1]
[2]

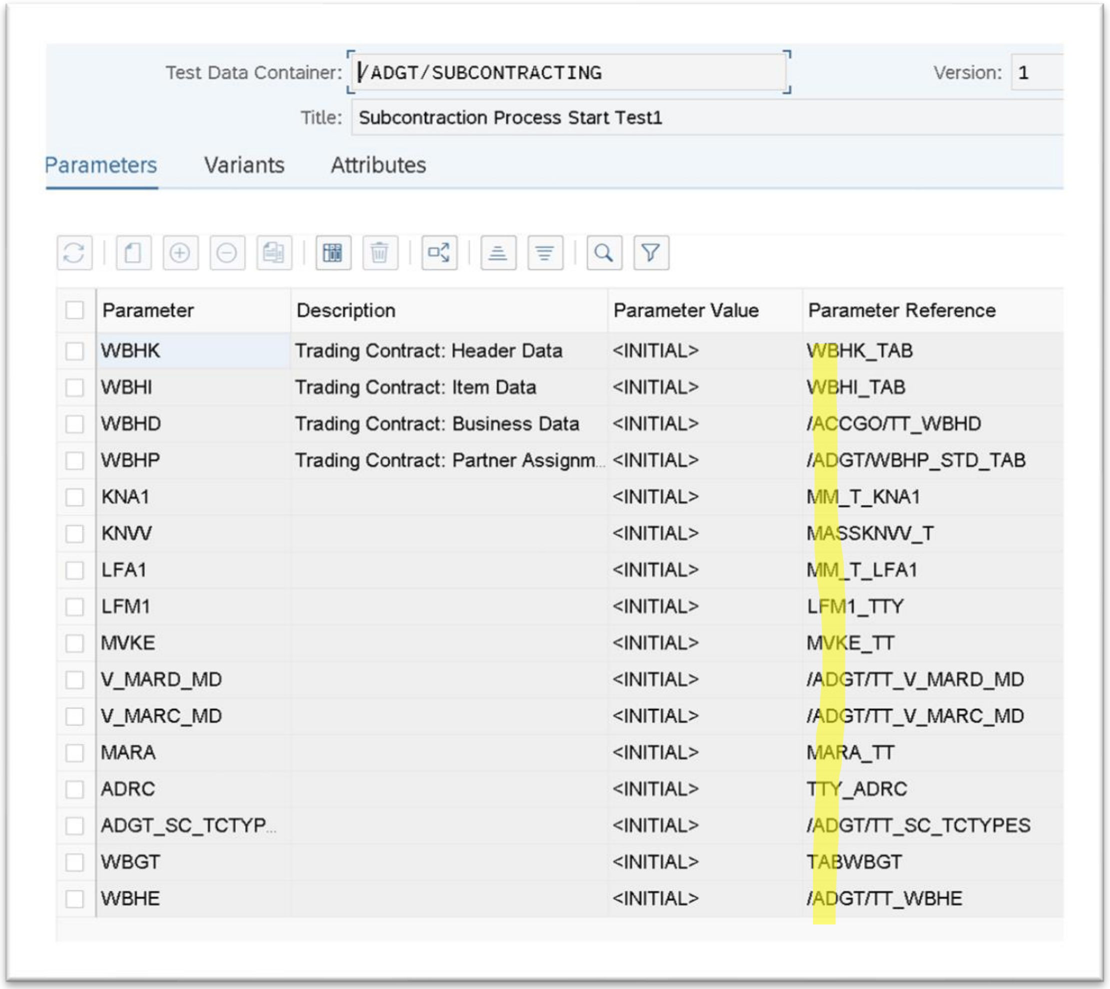
MANDT	(K)
MATNR	(K) 000000000000002124
WERKS	(K) 'NL01'
LGORT	(K) 1010
PSTAT	'D'
LVORM	
SPERR	
KZILL	
KZILQ	
KZILE	
KZILS	

V_MARD_MD(SUBCON_1)

MANDT	MATNR	WERKS	LGORT	PSTAT	LVORM	SPERR	KZILL
	000000000000002124	'NL01'	'SIT'	'D'			
	000000000000002124	'NL01'	1010	'D'			

ADventas Consulting

ECATT - Test data container

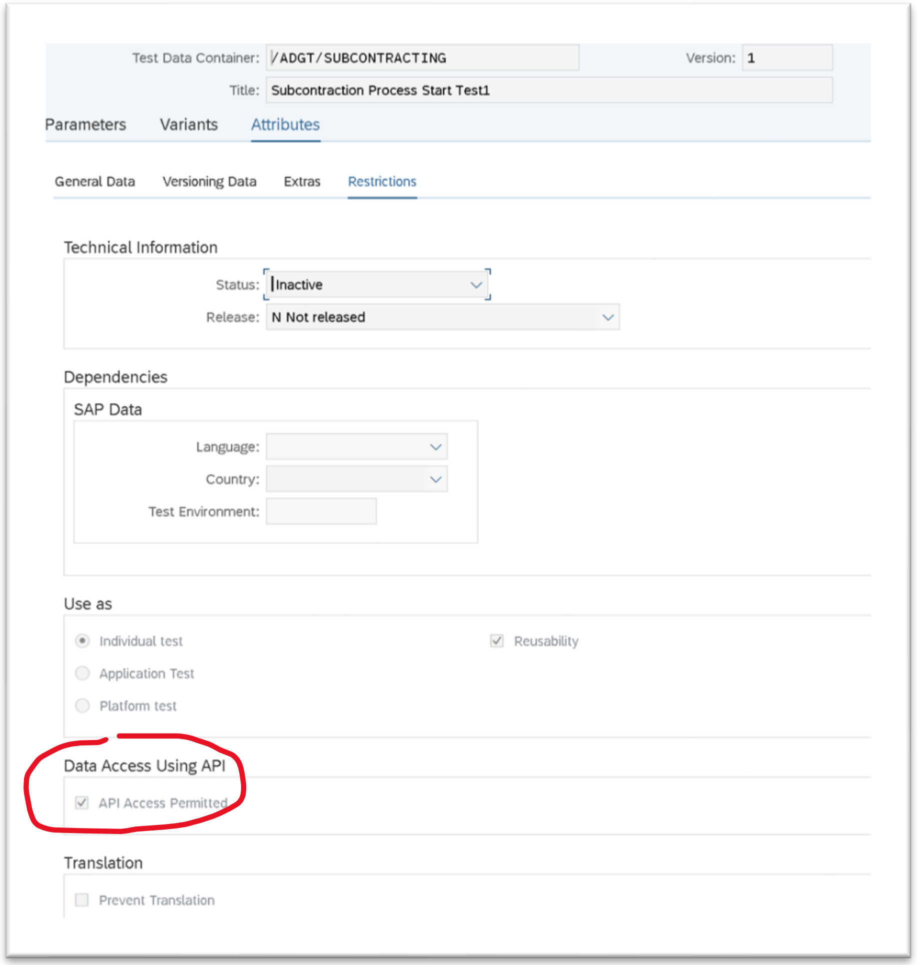


Test Data Container: /ADGT/SUBCONTRACTING   Version: 1

Title: Subcontraction Process Start Test1

Parameters | Variants | Attributes

Parameter	Description	Parameter Value	Parameter Reference
WBHK	Trading Contract: Header Data	<INITIAL>	WBHK_TAB
WBHI	Trading Contract: Item Data	<INITIAL>	WBHI_TAB
WBHD	Trading Contract: Business Data	<INITIAL>	/ACCGO/TT_WBHD
WBHP	Trading Contract: Partner Assignm...	<INITIAL>	/ADGT/WBHP_STD_TAB
KNA1		<INITIAL>	MM_T_KNA1
KNVV		<INITIAL>	MASSKNVV_T
LFA1		<INITIAL>	MM_T_LFA1
LFM1		<INITIAL>	LFM1_TTY
MVKE		<INITIAL>	MVKE_TT
V_MARD_MD		<INITIAL>	/ADGT/TT_V_MARD_MD
V_MARC_MD		<INITIAL>	/ADGT/TT_V_MARC_MD
MARA		<INITIAL>	MARA_TT
ADRC		<INITIAL>	TTY_ADRC
ADGT_SC_TCTYP...		<INITIAL>	/ADGT/TT_SC_TCTYPES
WBGT		<INITIAL>	TABWBGT
WBHE		<INITIAL>	/ADGT/TT_WBHE



Test Data Container: /ADGT/SUBCONTRACTING   Version: 1

Title: Subcontraction Process Start Test1

Parameters | Variants | Attributes

General Data | Versioning Data | Extras | Restrictions

Technical Information

Status: Inactive

Release: N Not released

Dependencies

SAP Data

Language:

Country:

Test Environment:

Use as

[● Individual test] ○ Application Test ○ Platform test

[✓ Reusability]

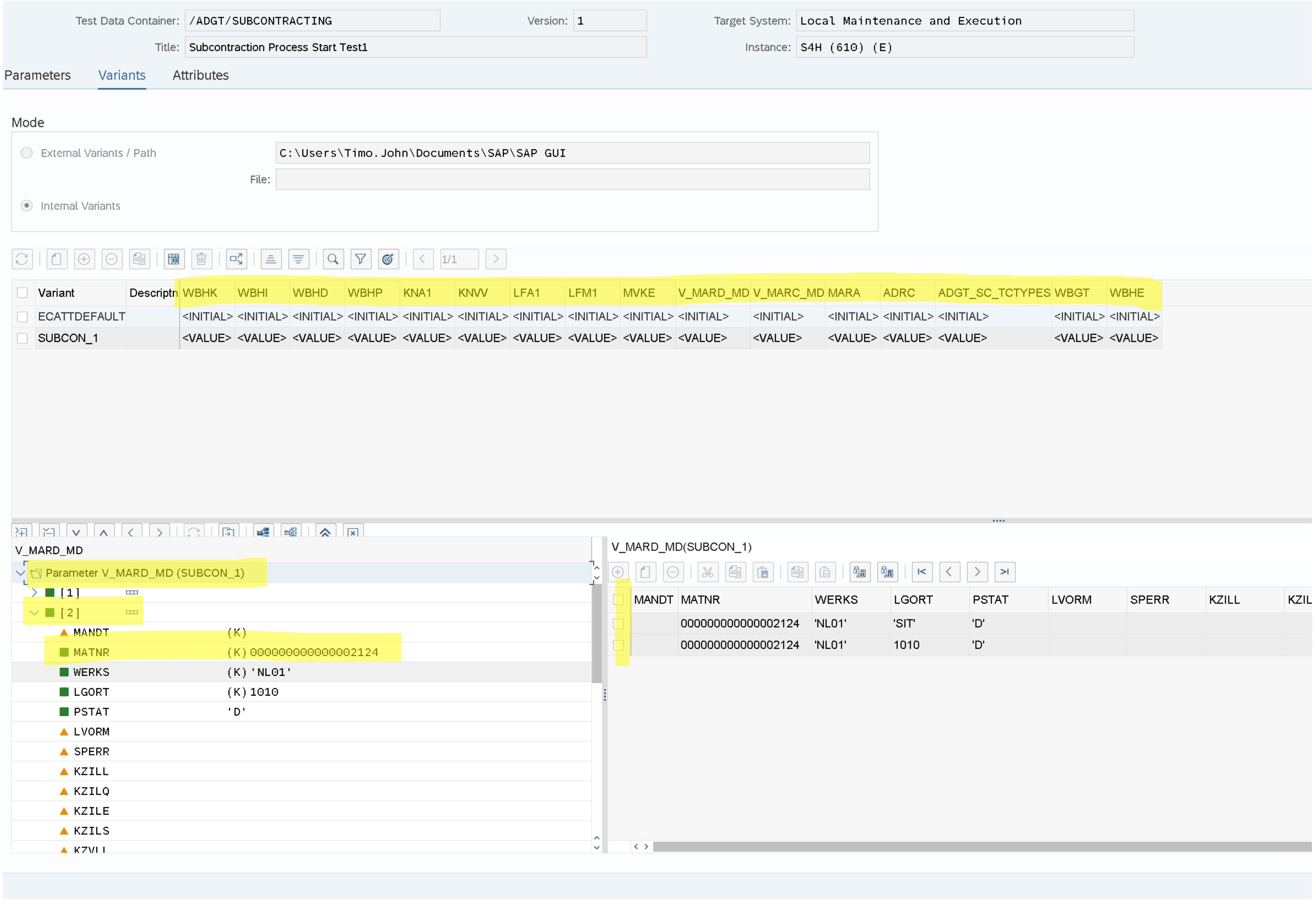
Data Access Using API

[✓ API Access Permitted]

Translation

☐ Prevent Translation

ADventas Consulting



Test Data Container: /ADGT/SUBCONTRACTING Version: 1 Target System: Local Maintenance and Execution

Title: Subcontraction Process Start Test1 Instance: S4H (610) (E)

Parameters | Variants | Attributes

Mode

External Variants / Path: C:\Users\Timo.John\Documents\SAP\SAP GUI

File:

Internal Variants

1/1

Variant	Descriptn	WBHK	WBHI	WBHD	WBHP	KNA1	KNVV	LFA1	LFM1	MVKE	V_MARD_MD	V_MARC_MD	MARA	ADRC	ADGT_SC_TCTYPES	WBGT	WBHE
ECATTDEFAULT		<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>	<INITIAL>
SUBCON_1		<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>	<VALUE>

V_MARD_MD

Parameter V_MARD_MD (SUBCON_1)

- [1]
- [2]

MANDT	(K)
MATNR	(K) 000000000000002124
WERKS	(K) 'NL01'
LGORT	(K) 1010
PSTAT	'D'
LVORM	
SPERR	
KZILL	
KZILQ	
KZILE	
KZILS	
KZVLL	

V_MARD_MD(SUBCON_1)

MANDT	MATNR	WERKS	LGORT	PSTAT	LVORM	SPERR	KZILL	KZIL
	000000000000002124	'NL01'	'SIT'	'D'				
	000000000000002124	'NL01'	1010	'D'				

Filling test data container

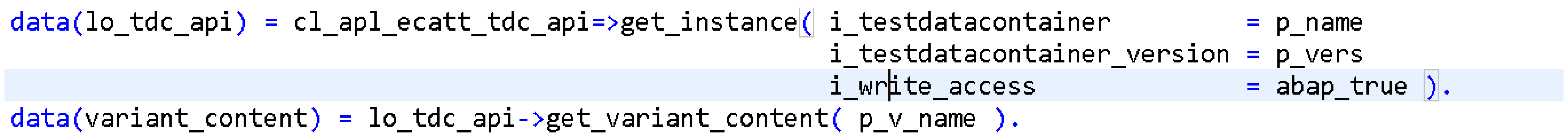
- Use a ABAP code to capture test date and write it into ECATT data container.
- A report could be executed in other clients / systems to extract test data into a TDC.

- Select needed full table lines of needed tables and store them in TDC parameter

- A little complex mapping of reference, but manageable.

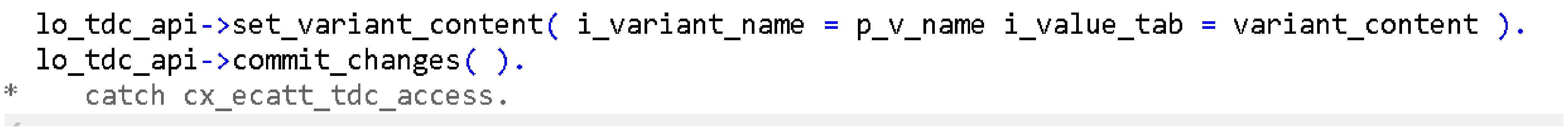
Filling test data container

- Read access in test class:



```
data(lo_tdc_api) = cl_apl_ecatt_tdc_api=>get_instance( i_testdatacontainer         = p_name
                                                       i_testdatacontainer_version = p_vers
                                                       i_write_access              = abap_true ).
data(variant_content) = lo_tdc_api->get_variant_content( p_v_name ).
```

- Write Access for TDC creation:



```
  lo_tdc_api->set_variant_content( i_variant_name = p_v_name i_value_tab = variant_content ).
  lo_tdc_api->commit_changes( ).
*    catch cx_ecatt_tdc_access.
```

ADventas Consulting

Connection ECATT TDC with Unit test Using “insert_from_tdc”:

```
" parameter name is name in the TDC and dependency the mocked DB object
" Important is that all mocked data do not have the source client
tdc_data_description = value #( tdc_name = c_ecatt_tdc_subcontraction tdc_variant = c_ecatt_tdc_variant_subcon_1 tdc_version = 1
                                ( tdc_parameter = 'WBHK'          dependency_name = 'WBHK' )
                                ( tdc_parameter = 'WBHD'          dependency_name = 'WBHD' )
                                ( tdc_parameter = 'WBHI'          dependency_name = 'WBHI' )
                                ( tdc_parameter = 'WBHP'          dependency_name = 'WBHP' )
                                ( tdc_parameter = 'WBHE'          dependency_name = 'WBHE' )
                                ( tdc_parameter = 'WBGT'          dependency_name = 'WBGT' )
                                ( tdc_parameter = 'KNA1'          dependency_name = 'KNA1' )
                                ( tdc_parameter = 'KNVV'          dependency_name = 'KNVV' )
                                ( tdc_parameter = 'LFA1'          dependency_name = 'LFA1' )
                                ( tdc_parameter = 'LFM1'          dependency_name = 'LFM1' )
                                ( tdc_parameter = 'MVKE'          dependency_name = 'MVKE' )
```

```
data(lo_db_mock) = cl_osql_test_environment=>create( value #( ( 'WBHK' ) ( 'WBHD' ) ( 'WBHI' ) ( 'WBHP' ) ( 'WBHE' ) ( 'WBGT' )
                                                              ( 'KNA1' ) ( 'KNVV' ) ( 'LFA1' ) ( 'LFM1' ) ( 'ADRC' )
                                                              ( 'MARA' ) ( 'MVKE' ) ( 'V_MARC_MD' ) ( 'V_MARD_MD' )
                                                              "Just Mock for write access needed:
                                                              ( 'WBHF' ) ( 'WBIT' ) ( 'WBCUMUL') ( 'KONV' )
                                                              ( '/ADGT/SC_TCTYPES') "#TODO 09.05.2022 ADVENTAS - Mock using Stub.
                                                              ( 'NRIV' ) ) ). " Number ranges.
```

```
lo_db_mock->insert_from_tdc( tdc_data_description ).
```

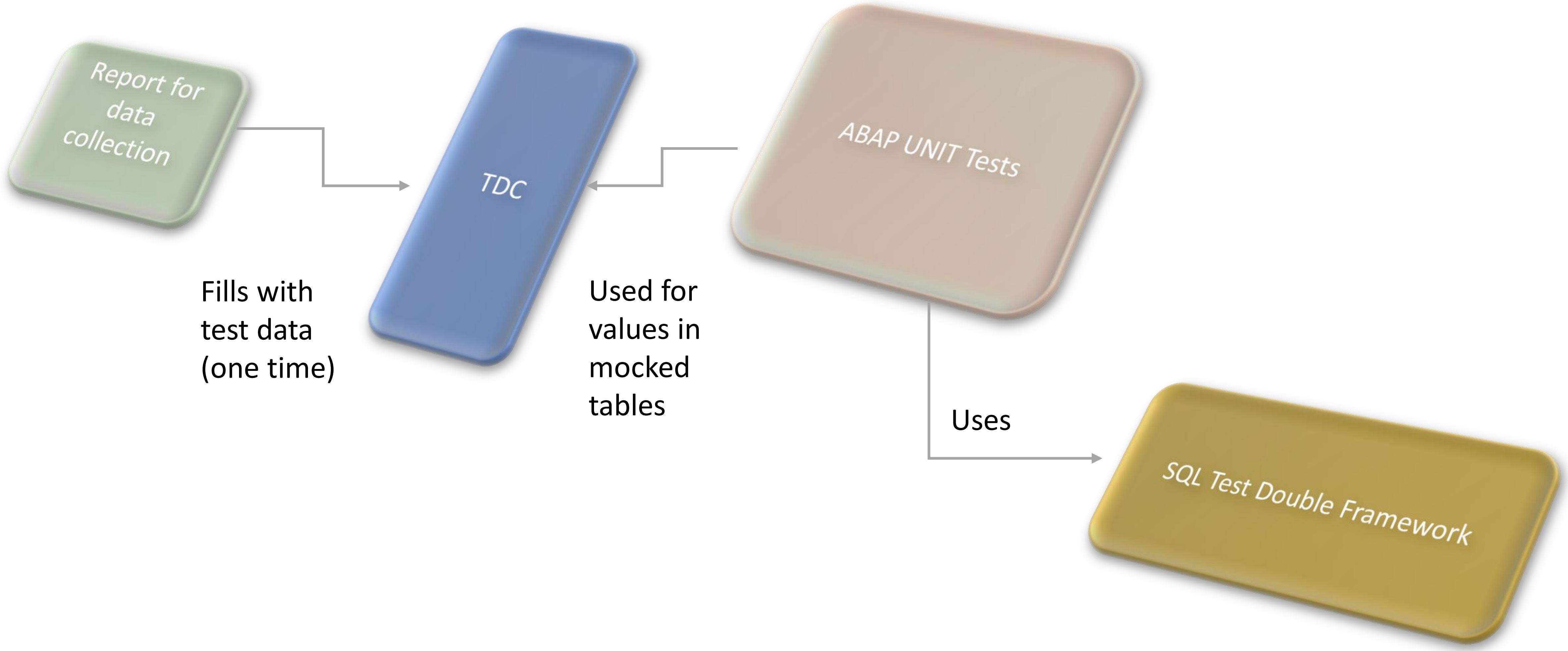
ADventas Consulting

Connection ECATT TDC with Unit test Using “insert_test_data”

```
data(lo_tdc_api) = cl_apl_ecatt_tdc_api=>get_instance( i_testdatacontainer         = c_ecatt_tdc_subcontrac
                                                        i_testdatacontainer_version = 1 ).
data(tdc_parameter_list) = lo_tdc_api->get_param_list( ).

"use parameter List of TDC and add some more tables that are needed in SAP code to write into
data(tables_to_be_mocked) = value #( base tdc_parameter_list
                                     ( 'WBHF' ) ( 'WBIT' ) ( 'WBCUMUL') ( 'KONV' )
                                     ( '/ADGT/SC_TCTYPES') "#TODO 09.05.2022 ADVENTAS - Mock using Stub.
                                     ( 'NRIV' ) ). " Number ranges.
delete tables_to_be_mocked where table_line = 'ADGT_SC_TCTYPES'. "beginning / is not allowed in TDC
data(lo_db_mock) = cl_osql_test_environment=>create( tables_to_be_mocked ).


"Fetch Test data from TDC:
data(tdc_variant_data) = lo_tdc_api->get_variant_content( i_variant_name = c_ecatt_tdc_variant_subcon_1 ).


"insert all test data from TDC into mock db.
loop at tdc_variant_data into data(variant_data).
  assign variant_data-value_ref->* to <testdata>.
  lo_db_mock->insert_test_data( i_data = <testdata> ).
endloop.

commit work.✓
```

Faster than “insert from_tdc”

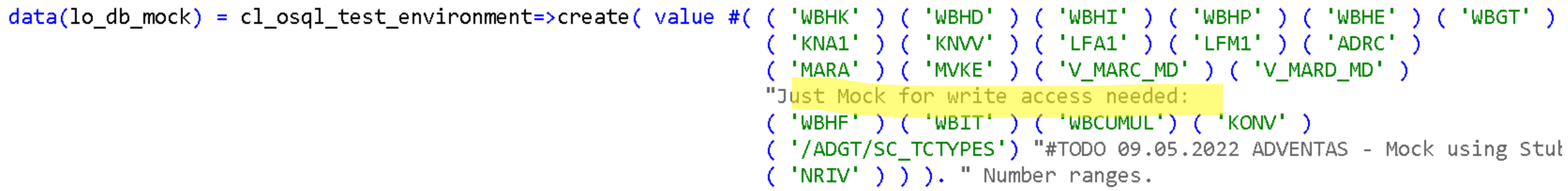
Big Picture – Recap


Report for data collection
Fills with test data (one time)
TDC
Used for values in mocked tables
ABAP UNIT Tests
Uses
SQL Test Double Framework

ADventas Consulting

Hints

- Mock tables where designated records are needed for the test.
  - Can be master data example vendor master
  - transitional data.
- Do NOT create mock data for each table!
  - Example T100 makes no sense to mock
- Tables not mocked will be available from regular DB.



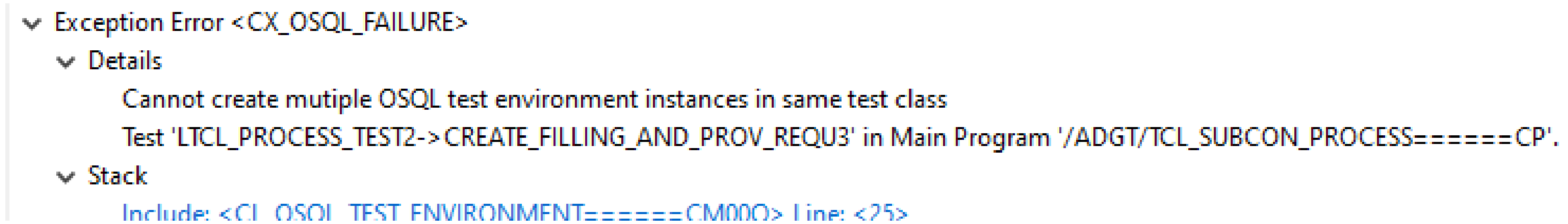
```
data(lo_db_mock) = cl_osql_test_environment=>create( value #( ( 'WBHK' ) ( 'WBHD' ) ( 'WBHI' ) ( 'WBHP' ) ( 'WBHE' ) ( 'WBGT' )
                                                              ( 'KNA1' ) ( 'KNVV' ) ( 'LFA1' ) ( 'LFM1' ) ( 'ADRC' )
                                                              ( 'MARA' ) ( 'MVKE' ) ( 'V_MARC_MD' ) ( 'V_MARD_MD' )
                                                              "Just Mock for write access needed:
                                                              ( 'WBHF' ) ( 'WBIT' ) ( 'WBCUMUL') ( 'KONV' )
                                                              ( '/ADGT/SC_TCTYPES') "#TODO 09.05.2022 ADVENTAS - Mock using Stu
                                                              ( 'NRIV' ) ) ). " Number ranges.
```

ADventas Consulting

Hints

- Attention with using parallel processes here the regular DB is used

- Create mock in class-setup!



```
Exception Error <CX_OSQL_FAILURE>
  Details
    Cannot create mutiple OSQL test environment instances in same test class
    Test 'LTCL_PROCESS_TEST2->CREATE_FILLING_AND_PROV_REQU3' in Main Program '/ADGT/TCL_SUBCON_PROCESS======CP'.
  Stack
    Include: <CL_OSQL_TEST_ENVIRONMENT======CM00O> Line: <25>
```

Thank you for your attention and interest

Cheat sheet

```
method class_setup.
"can be run only once per test
class.
lo_db_mock =
cl_osql_test_environment=>create(
value #( ( 'T100' ) ) ). "

"time consuming prepare of test data
store in class attribute for reuse.
t100_mock = value t100( msgnr =
'002' arbgb = '00' sprsl = 'E' text
= 'unit test demo' ).
endmethod.
```

```
method setup.
   " Remove old Test data.
   lo_db_mock->clear_doubles( ).

   "Insert fresh test data
   insert t100 from t100_mock .
   cut = new #(  ).
 endmethod.


 method class_teardown.
   lo_db_mock->destroy( ).
 endmethod.
```

Cheat sheet

- TA: SECATT

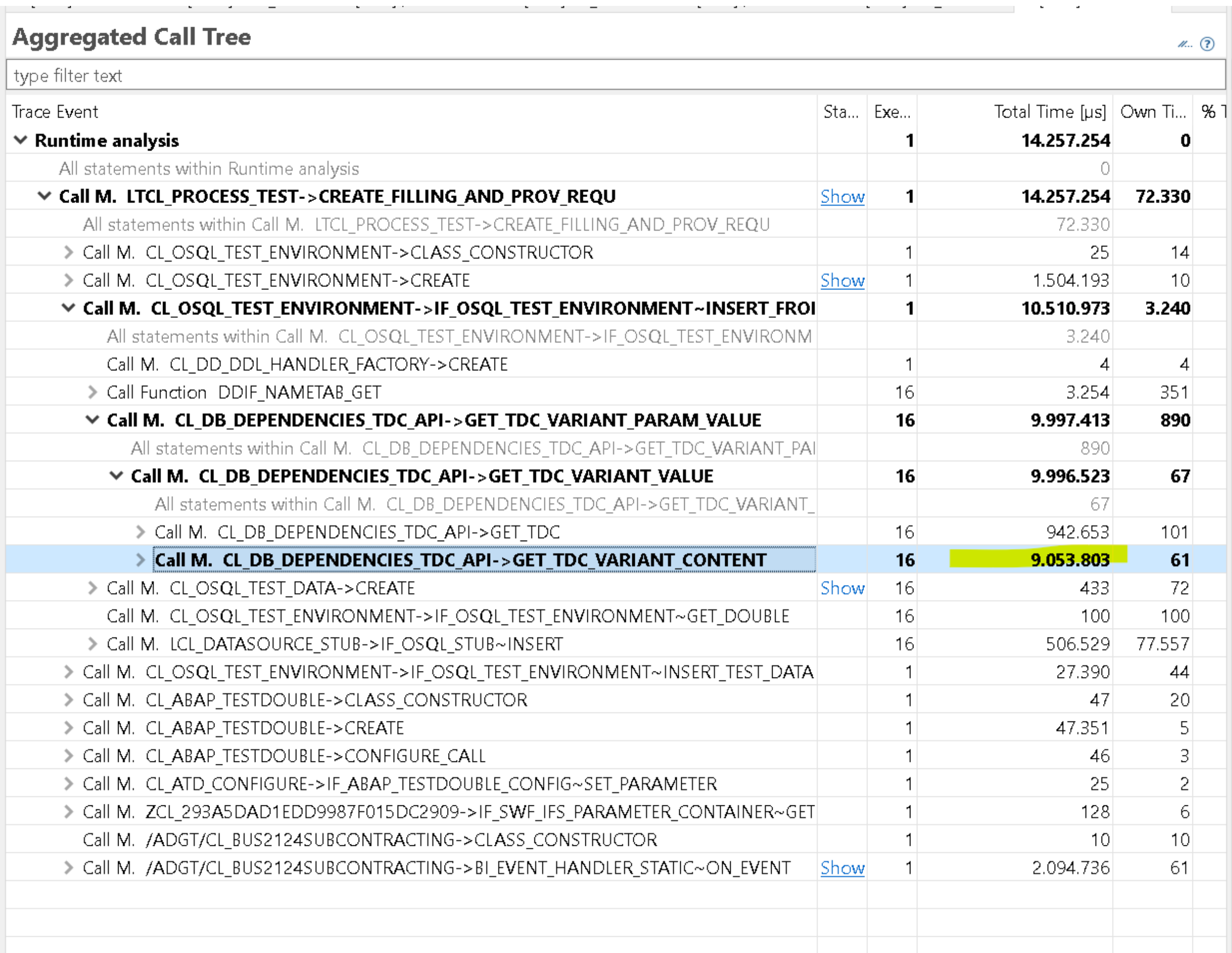
```
•  lo_tdc_api->set_variant_content(
  i_variant_name = p_v_name
  i_value_tab = variant_content ).
•   lo_tdc_api->commit_changes( ).
```

```
• data(tdc_variant_data) =
  lo_tdc_api->get_variant_content(
  i_variant_name = …
```

```
• data(lo_db_mock) =
  cl_osql_test_environment=>create(
  tables_to_be_mocked ).
• lo_db_mock->insert_test_data( i_data =
  <testdata> ).
• data(tdc_variant_data) = lo_tdc_api-
  >get_variant_content( i_variant_name =
```

ADventas Consulting

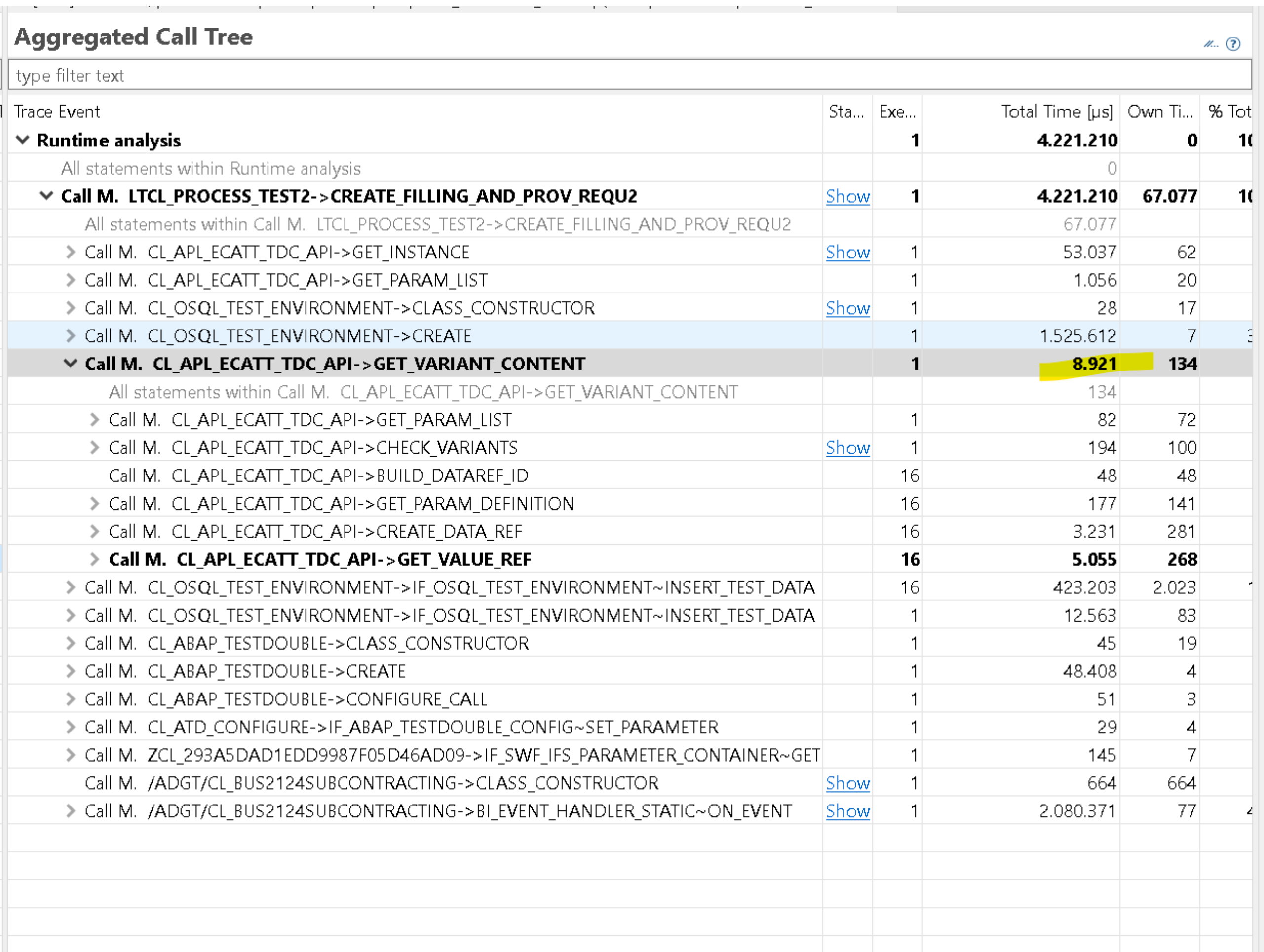
Why is the SAP API much slower?



Aggregated Call Tree

type filter text

Trace Event	Sta...	Exe...	Total Time [µs]	Own Ti...	% T
Runtime analysis		**1**	**14.257.254**	**0**	
All statements within Runtime analysis			0		
Call M. LTCL_PROCESS_TEST->CREATE_FILLING_AND_PROV_REQU	Show	**1**	**14.257.254**	**72.330**	
All statements within Call M. LTCL_PROCESS_TEST->CREATE_FILLING_AND_PROV_REQU			72.330		
Call M. CL_OSQL_TEST_ENVIRONMENT->CLASS_CONSTRUCTOR		1	25	14	
Call M. CL_OSQL_TEST_ENVIRONMENT->CREATE	Show	1	1.504.193	10	
Call M. CL_OSQL_TEST_ENVIRONMENT->IF_OSQL_TEST_ENVIRONMENT~INSERT_FROI		**1**	**10.510.973**	**3.240**	
All statements within Call M. CL_OSQL_TEST_ENVIRONMENT->IF_OSQL_TEST_ENVIRONM			3.240		
Call M. CL_DD_DDL_HANDLER_FACTORY->CREATE		1	4	4	
Call Function DDIF_NAMETAB_GET		16	3.254	351	
Call M. CL_DB_DEPENDENCIES_TDC_API->GET_TDC_VARIANT_PARAM_VALUE		**16**	**9.997.413**	**890**	
All statements within Call M. CL_DB_DEPENDENCIES_TDC_API->GET_TDC_VARIANT_PAI			890		
Call M. CL_DB_DEPENDENCIES_TDC_API->GET_TDC_VARIANT_VALUE		**16**	**9.996.523**	**67**	
All statements within Call M. CL_DB_DEPENDENCIES_TDC_API->GET_TDC_VARIANT_			67		
Call M. CL_DB_DEPENDENCIES_TDC_API->GET_TDC		16	942.653	101	
Call M. CL_DB_DEPENDENCIES_TDC_API->GET_TDC_VARIANT_CONTENT		**16**	**9.053.803**	**61**	
Call M. CL_OSQL_TEST_DATA->CREATE	Show	16	433	72	
Call M. CL_OSQL_TEST_ENVIRONMENT->IF_OSQL_TEST_ENVIRONMENT~GET_DOUBLE		16	100	100	
Call M. LCL_DATASOURCE_STUB->IF_OSQL_STUB~INSERT		16	506.529	77.557	
Call M. CL_OSQL_TEST_ENVIRONMENT->IF_OSQL_TEST_ENVIRONMENT~INSERT_TEST_DATA		1	27.390	44	
Call M. CL_ABAP_TESTDOUBLE->CLASS_CONSTRUCTOR		1	47	20	
Call M. CL_ABAP_TESTDOUBLE->CREATE		1	47.351	5	
Call M. CL_ABAP_TESTDOUBLE->CONFIGURE_CALL		1	46	3	
Call M. CL_ATD_CONFIGURE->IF_ABAP_TESTDOUBLE_CONFIG~SET_PARAMETER		1	25	2	
Call M. ZCL_293A5DAD1EDD9987F015DC2909->IF_SWF_IFS_PARAMETER_CONTAINER~GET		1	128	6	
Call M. /ADGT/CL_BUS2124SUBCONTRACTING->CLASS_CONSTRUCTOR		1	10	10	
Call M. /ADGT/CL_BUS2124SUBCONTRACTING->BI_EVENT_HANDLER_STATIC~ON_EVENT	Show	1	2.094.736	61	



Aggregated Call Tree

type filter text

Trace Event	Sta...	Exe...	Total Time [µs]	Own Ti...	% Tot
Runtime analysis		**1**	**4.221.210**	**0**	**10**
All statements within Runtime analysis			0		
Call M. LTCL_PROCESS_TEST2->CREATE_FILLING_AND_PROV_REQU2	Show	**1**	**4.221.210**	**67.077**	**10**
All statements within Call M. LTCL_PROCESS_TEST2->CREATE_FILLING_AND_PROV_REQU2			67.077		
Call M. CL_APL_ECATT_TDC_API->GET_INSTANCE	Show	1	53.037	62	
Call M. CL_APL_ECATT_TDC_API->GET_PARAM_LIST		1	1.056	20	
Call M. CL_OSQL_TEST_ENVIRONMENT->CLASS_CONSTRUCTOR	Show	1	28	17	
Call M. CL_OSQL_TEST_ENVIRONMENT->CREATE		1	1.525.612	7	3
Call M. CL_APL_ECATT_TDC_API->GET_VARIANT_CONTENT		**1**	**8.921**	**134**	
All statements within Call M. CL_APL_ECATT_TDC_API->GET_VARIANT_CONTENT			134		
Call M. CL_APL_ECATT_TDC_API->GET_PARAM_LIST		1	82	72	
Call M. CL_APL_ECATT_TDC_API->CHECK_VARIANTS	Show	1	194	100	
Call M. CL_APL_ECATT_TDC_API->BUILD_DATAREF_ID		16	48	48	
Call M. CL_APL_ECATT_TDC_API->GET_PARAM_DEFINITION		16	177	141	
Call M. CL_APL_ECATT_TDC_API->CREATE_DATA_REF		16	3.231	281	
Call M. CL_APL_ECATT_TDC_API->GET_VALUE_REF		**16**	**5.055**	**268**	
Call M. CL_OSQL_TEST_ENVIRONMENT->IF_OSQL_TEST_ENVIRONMENT~INSERT_TEST_DATA		16	423.203	2.023	1
Call M. CL_OSQL_TEST_ENVIRONMENT->IF_OSQL_TEST_ENVIRONMENT~INSERT_TEST_DATA		1	12.563	83	
Call M. CL_ABAP_TESTDOUBLE->CLASS_CONSTRUCTOR		1	45	19	
Call M. CL_ABAP_TESTDOUBLE->CREATE		1	48.408	4	
Call M. CL_ABAP_TESTDOUBLE->CONFIGURE_CALL		1	51	3	
Call M. CL_ATD_CONFIGURE->IF_ABAP_TESTDOUBLE_CONFIG~SET_PARAMETER		1	29	4	
Call M. ZCL_293A5DAD1EDD9987F05D46AD09->IF_SWF_IFS_PARAMETER_CONTAINER~GET		1	145	7	
Call M. /ADGT/CL_BUS2124SUBCONTRACTING->CLASS_CONSTRUCTOR	Show	1	664	664	
Call M. /ADGT/CL_BUS2124SUBCONTRACTING->BI_EVENT_HANDLER_STATIC~ON_EVENT	Show	1	2.080.371	77	4

There

ADventas Consulting

Resources

- SAP Help Portal : SAP (On-Premise) - ABAP Development User Guide https://help.sap.com/docs/ABAP_PLATFORM/c238d694b825421f940829321ffa326a/8562b437073d4b9c93078c45f7a64f21.html?locale=en-US(Accessed 14.11.2022)

- opensap.com : Writing Testable Code for ABAP 2018

ADventas Consulting