Uwe Kunath
January 11, 2014 3 minute read

Enqueues and Update Tasks

What help.sap.com doesn't tell us

I thought, SAP is pretty clear in what they tell us about the _SCOPE-parameter that can be passed to every enqueue-function modul.

The parameter can have 3 values, which means, according to help.sap.com:

Meaning of the _SCOPE Values

Value	Description
_SCOPE = 1	The lock belongs only to the dialog owner (owner_1), and therefore only exists in the dialog transaction. The `DEQUEUE` call or the end of the transaction, not `COMMIT WORK` or `ROLLBACK WORK`,cancels the lock.
_SCOPE = 2	The lock belongs to the update owner (owner_2) only. Therefore,

	the update inherits the lock when `CALL FUNCTION '...' IN UPDATE TASK` and `COMMIT WORK` are called. The lock is released when the update transaction is complete. You can release the lock before it is transferred to the update using `ROLLBACK WORK`. `COMMIT WORK` has no effect, unless `CALL FUNCTION '...' IN UPDATE TASK` has been called.
_SCOPE = 3	The lock belongs to both owners (owner_1 and owner_2). In other words, it combines the behavior of both. This lock is canceled when the last of the two owners has released it.

_SCOPE Parameter (SAP Library – Architecture Manual)

The most interesting statement is the description for _SCOPE = 2. It says that the lock is released when the update transaction is complete.

So I understood that COMMIT WORK will implicitly release the lock. As I found out, it is not necessarily the case.

The test report

Consider the following test report.

The report is going to request a lock for an ABAP report. The lock argument is '1' as a dummy value. Keep in mind, that Z_UPDATE_TASK_DUMMY is an update function module that has no execution logic in my case.

```
DATA lt_enq TYPE TABLE OF seqg3.
FIELD-SYMBOLS <ls_enq> TYPE seqg3.
PARAMETERS pa_upd TYPE abap_bool AS CHECKBOX.
CALL FUNCTION 'ENQUEUE_ES_PROG'
   EXPORTING
     name           = '1'
     _scope         = '2'
   EXCEPTIONS
     foreign_lock   = 1
     system_failure = 2
```

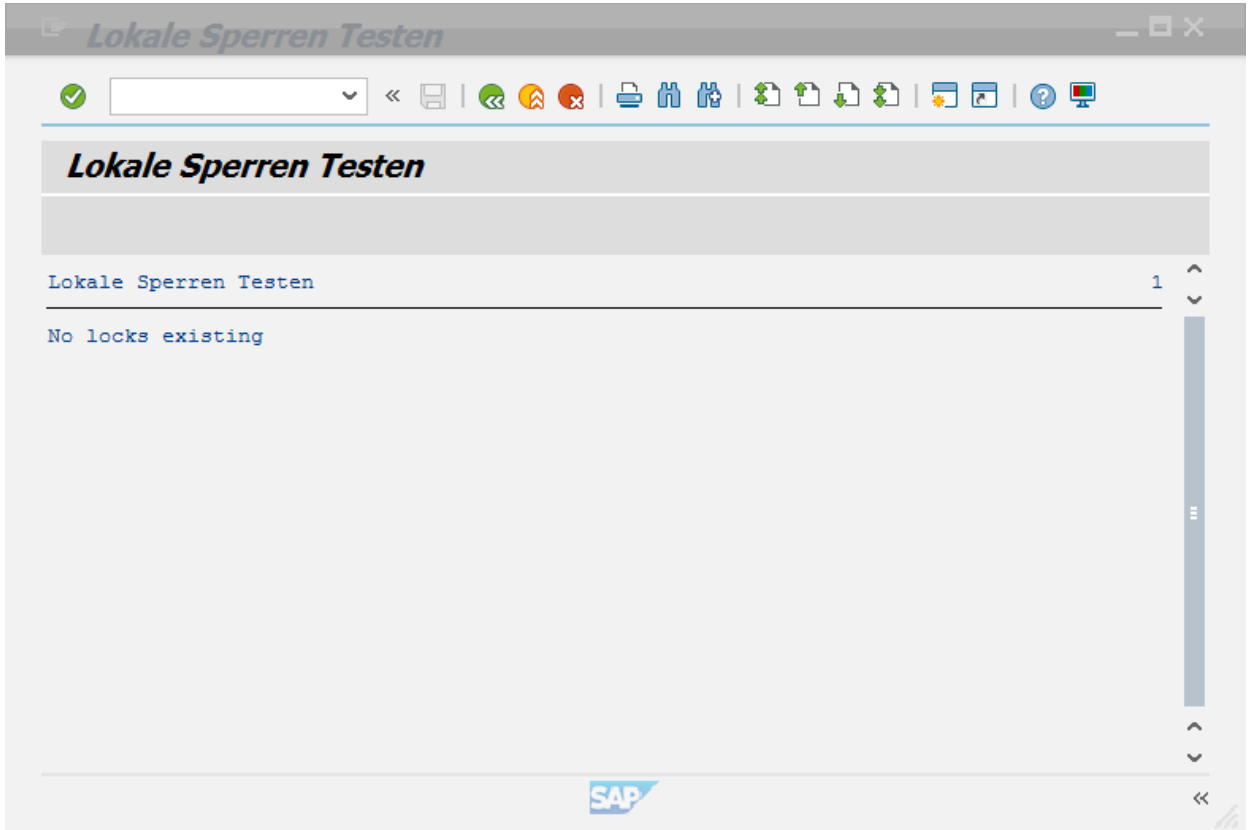
```
      OTHERS         = 3.
IF sy-subrc <> 0.
   MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ENDIF.
IF pa_upd = abap_true.
   CALL FUNCTION 'Z_UPDATE_TASK_DUMMY' IN UPDATE TASK.
ENDIF.
COMMIT WORK AND WAIT.
CALL FUNCTION 'ENQUEUE_READ'
   EXPORTING
     gclient               = sy-mandt
     guname                = sy-uname
   TABLES
     enq                   = lt_enq
   EXCEPTIONS
     communication_failure = 1
     system_failure        = 2
     OTHERS                = 3.
IF sy-subrc <> 0.
   MESSAGE ID sy-msgid TYPE sy-msgty NUMBER sy-msgno WITH sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4.
ELSE.
   IF lt_enq IS INITIAL.
     WRITE 'No locks existing'.
   ELSE.
     LOOP AT lt_enq ASSIGNING <ls_enq>.
       WRITE: <ls_enq>-gname,
              <ls_enq>-garg,
              <ls_enq>-gmode.
       NEW-LINE.
     ENDLOOP.
   ENDIF.
ENDIF.
```

The report offers a parameter PA_UPD to the user. If checked, the update function module will be called. If not checked, there is no update task registered. Afterwards, COMMIT WORK AND WAIT is always triggered.

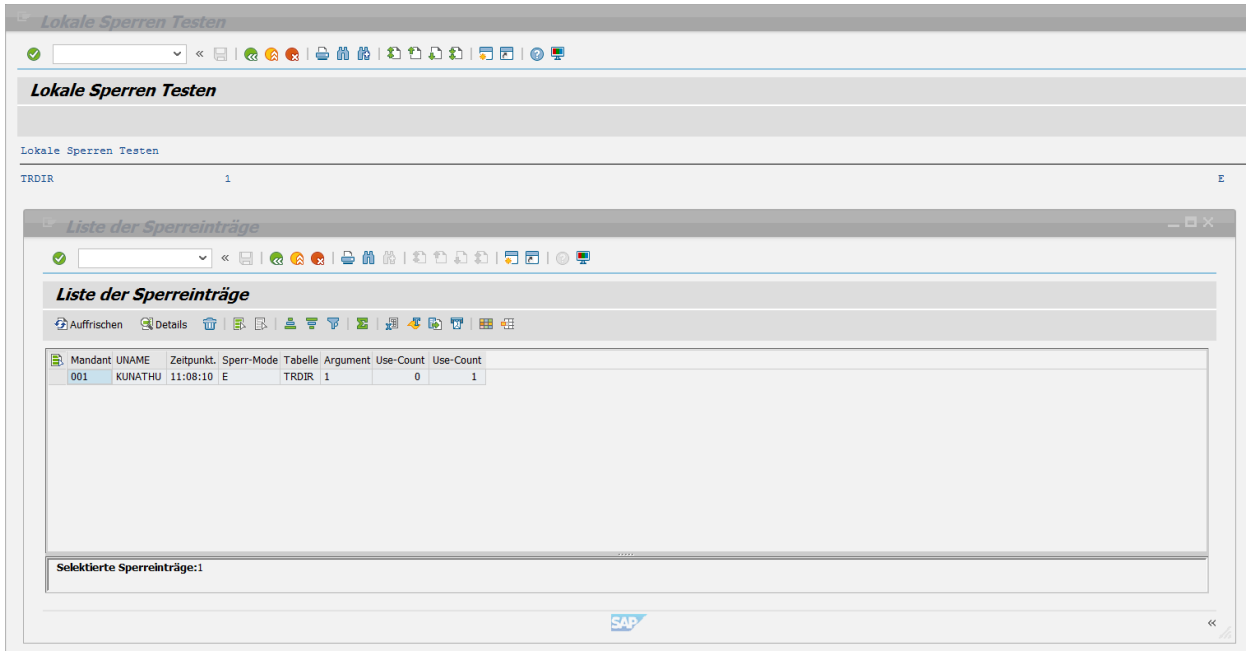
In the end, the report shows any locks that the current user has.

The ugly truth

If the report is called with the enabled checkbox, the dummy function module will be called. In the end, all locks are released.



If the report should not execute any update function module (checkbox is deactivated), the locks are not released at all. COMMIT WORK AND WAIT has no influence here as there is no update task registered. I also included the output of SM12 in the screenshot.



The moral of the story

Locks of the update owner are not necessarily released, when COMMIT WORK has been called. It depends on wether or not an update task has been called at all. help.sap.com is not wrong with the description about _SCOPE = 2, however, someone who reads the description might get a wrong idea of how ABAP works in this very case. It took me some time to figure out what happens here. So I hope this blog post helps other developers saving some time.

Alert Moderator

Assigned tags

ABAP Development | abap | commit | enqueue | update task |

Related Blog Posts

ADT Basics: Transport Organizer
By **Raghuvira BHAGAVAN** , Feb 20, 2013

ABAP in eclipse – team work with tasks
By **Adam Krawczyk** , Aug 05, 2016

My own Function-Module to calculate duration between 2 dates and 2 times
By **Jörg Sauterleute** , Jan 16, 2016

Related Questions

ENQUEUEs and COMMITs
By **Former Member** , Aug 28, 2007

Deadlock with Enqueue Module
By **Sunil K Achyut** , Nov 04, 2005

Enqueue and Dequeue AUSP
By **Former Member** , Feb 01, 2011

3 Comments

You must be Logged on to comment or reply to a post.

Suhas Saha

January 12, 2014 at 11:29 am

Hello Uwe,

As per the SAP documentation for *_SCOPE = '2'*:

> *COMMIT WORK has no effect, unless CALL FUNCTION '...' IN UPDATE TASK has been called.*

What you have found out:

> *Locks of the update owner are not necessarily released, when COMMIT WORK has been called. It depends on wether or not an update task has been called at all.*

I think both are the same, aren't they?

Cheers,

Suhas

Like(0)

Uwe Kunath | Post author

January 12, 2014 at 12:09 pm

Dear Suhas,

yes you are right. I just misunderstood this statement in a sense that no DB update takes place. How silly 🙂

However, as I wrote, help.sap.com is not wrong, I just misunderstood the described behaviour. So I hope this blog post is still helpful for some developers…

Regards,

Uwe

Like(0)

Marcin Makowski

February 19, 2020 at 12:46 pm

Thank you Uwe!

it became useful to me right now in 2020 🙂 Cheers!

BR,

Marcin