



Former Member

December 19, 2012 | 3 minute read

Application Logging using ABAP

💬 3 👍 4 👁 29,722

It's been always a headache when customer raises a message and you are not sure what might have gone wrong!!!

At this point if you have application logging in place it can be really helpful to trace and check which code might have triggered the issue.

Application logging can be useful for the below scenarios too:

- 1...When you want to record the progress of the execution of an application so that you can reconstruct it later if necessary
- 2...when you are not sure/not able to debug a code (foreground/background) and you want to drill down the error cause. In this case you can easily check the application log and decide the exact location where the problem can be.

So let's start how you can create an application logging.

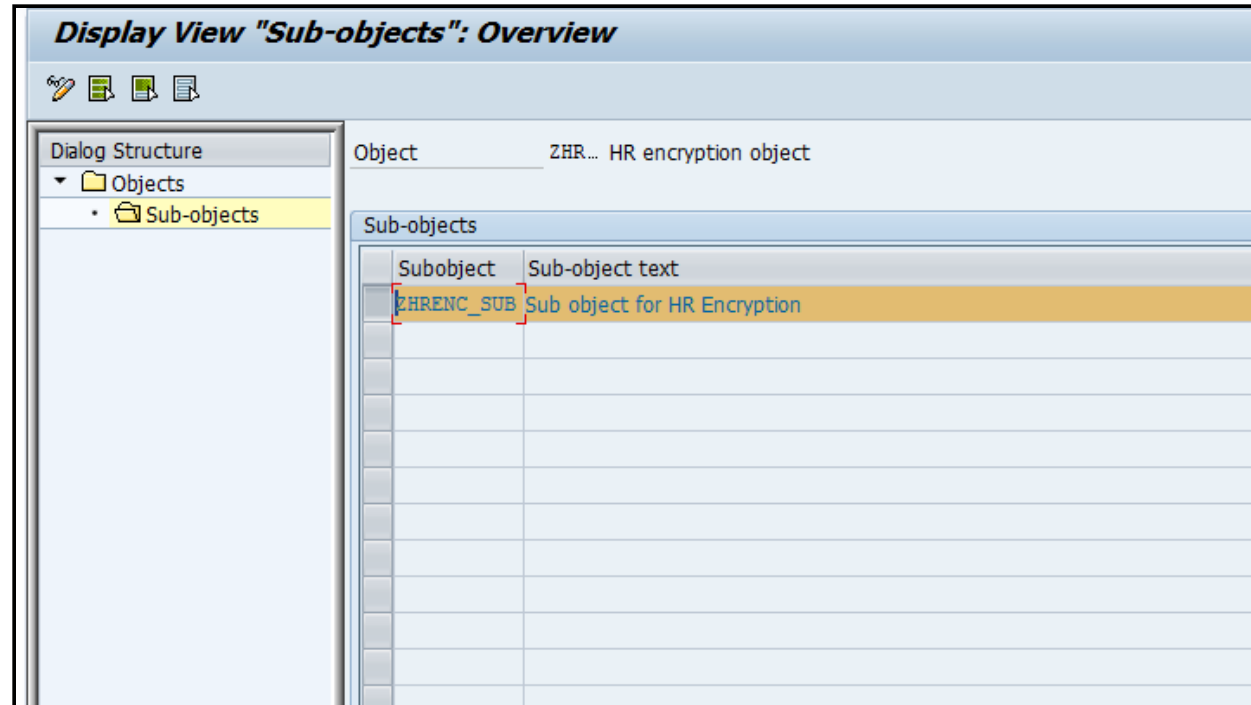
Transaction code: SLG0 is used to create an object which is used to identify your logs.

So here I have created an application log Object named: ZHR_ENC.

[illegible]

And a Sub-object ZHRENC_SUB.

Its like: I want to create an application log for COUNTRY management. And inside the countries whenever application is related to a particular state. I create sub-object based on the states.



Once you are done with the creation of the Object and sub object it's time to put the code which will actually create the application logs.

You can write the below code to your enhancements/User-Exit/reports (any custom code) so that when it is executed logging is done.

(For testing purpose just copy the below code in Report program **ZTST_APPL_LOG**)

```

*&-----*
*& Report ZTST_APPL_LOG
*&
*&-----*
*&
*&
*&-----*

```

REPORT ZTST_APPL_LOG.

type-pools: abap.

*data: l_log_handle type balloghndl,
l_timestamp type tzntstmps,
l_timezone type timezone value 'UTC',
l_str_log type bal_s_log,
l_str_balmsg type bal_s_msg,
l_str_message type bapiret2,
l_msg_logged type boolean,
l_tab_messages type bapiret2_t.*

**-Building messages*

**-Use your own message which you want to Display in the log
do 1 times.*

call function 'BALW_BAPIRETURN_GET2'

exporting

type = 'E'

cl = 'BPFS'

number = '006'

importing

return = l_str_message.

append l_str_message to l_tab_messages.

clear l_str_message.

enddo.

**-Logging messages*

convert date sy-datum time sy-uzeit

into time stamp l_timestamp time zone l_timezone.

l_str_log-extnumber = l_timestamp.

condense l_str_log-extnumber.

l_str_log-object = 'ZHR_ENC'.

l_str_log-subobject = 'ZHRENC_SUB'.

l_str_log-aldate_del = sy-datum + 5.

call function 'BAL_LOG_CREATE'

exporting

```

i_s_log = l_str_log
importing
e_log_handle = l_log_handle
exceptions
log_header_inconsistent = 1
others = 2.
if sy-subrc <> 0.
message id sy-msgid type sy-msgty number sy-msgno
with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 into l_str_message-message.
write: 'Type',sy-msgty, 'Message',l_str_message-message.
else.
loop at l_tab_messages into l_str_message.
move: l_str_message-type    to l_str_balmsg-msgty,
l_str_message-id          to l_str_balmsg-msgid,
l_str_message-number      to l_str_balmsg-msgno,
l_str_message-message_v1 to l_str_balmsg-msgv1,
l_str_message-message_v2 to l_str_balmsg-msgv2,
l_str_message-message_v3 to l_str_balmsg-msgv3,
l_str_message-message_v4 to l_str_balmsg-msgv4.

call function 'BAL_LOG_MSG_ADD'
exporting
i_log_handle = l_log_handle
i_s_msg = l_str_balmsg
importing
e_msg_was_logged = l_msg_logged
exceptions
log_not_found = 1
msg_inconsistent = 2
log_is_full = 3
others = 4.
if sy-subrc <> 0.
message id sy-msgid type sy-msgty number sy-msgno
with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 into l_str_message-message.
write: 'Type',sy-msgty, 'Message',l_str_message-message.

```

```

endif.
endloop.
if sy-subrc eq 0.
call function 'BAL_DB_SAVE'
exporting
i_save_all = abap_true
exceptions
log_not_found = 1
save_not_allowed = 2
numbering_error = 3
others = 4.
if sy-subrc <> 0.
message id sy-msgid type sy-msgty number sy-msgno
with sy-msgv1 sy-msgv2 sy-msgv3 sy-msgv4 into l_str_message-message.
write: 'Type',sy-msgty, 'Message',l_str_message-message.
else.
write: 'Messages Saved in the log'.
endif.
endif.
endif.
write : 'done with log number',l_str_log-extnumber.

```

In the above code three function modules are doing the full job.

'BAL_LOG_CREATE' it creates the application logging object

'BAL_LOG_MSG_ADD' it adds the messages to your application logging object/sub-object

'BAL_DB_SAVE' saves the messages into to database.

Note: Here I have used the standard message class BPFS (message number: 006).Incase if it's not available in your system create new one using Transaction code SE91 or reuse any existing message.

Execute the created report .You can use the transaction Code: SLG1 to monitor the application logging based on the object.

Analyse Application Log

Object	ZHR_ENC	
Subobject	*	
External ID	*	

Time Restriction

From (Date/Time)	31.08.2012		11:35:05	
To (Date/Time)	31.08.2012		23:59:59	

Log Triggered By

User	*	
Transaction code	*	
Program	*	

Log Class	Log Creation
<input type="radio"/> Only very important logs <input type="radio"/> Only important logs <input type="radio"/> Also less important logs <input checked="" type="radio"/> All logs	<input checked="" type="radio"/> Any <input type="radio"/> Dialog <input type="radio"/> In batch mode <input type="radio"/> Batch input

Log Source and Formatting

☒ Format Completely from Database
☐ Format Only Header Data from Database
☐ Format Completely from Archive

So now when customers raise the Ticket the first step you can do is. Check the application logging in the production and you will have clear picture what might have been gone wrong.

Follow



Like



RSS Feed

Display logs								
Technical Information								
Date/Time/User	Nu...	External ID	Object text	Sub-object text	Transac...	Program	Mode	Log number
31.08.2012 11:35:16 DEV01	1	201208311135...	HR encryption ...	Sub object for ...	SE38		Dialog pro...	00000000000005851476
• Problem class Other	1							
.....								
T... Message Text								
Enter the business partner number								

Alert Moderator

Assigned Tags

ABAP Development

abap

Similar Blog Posts

[Business Application Log \(BAL\) - Message Statistics](#)



By Emanuel Klenner Jan 11, 2022

[SAP ABAP Platform 1909, Developer Edition: AVAILABLE NOW](#)

By Julie Plummer Feb 15, 2021

[Getting Started with the ABAP RESTful Application Programming Model \(RAP\)](#)

By Carine Tchoutouo Djomo Oct 25, 2019

Related Questions

[Application Logging in ABAP](#)

By Sushil TANEJA Mar 08, 2010

[ABAP Application log](#)

By Former Member May 13, 2009

[what are F. M's used for creating application log in application server](#)

By Former Member Aug 29, 2008

Join the Conversation



[SAP TechEd](#)

Tune in for tech talk. Stay for inspiration. Upskill your future.



[SAP BTP Learning Group](#)

SAP Business Technology Platform Learning Journeys.

[Coffee Corner](#)



Join the new Coffee Corner Discussion Group.

3 Comments

You must be [Logged on](#) to comment or reply to a post.



Former Member

June 6, 2013 at 12:29 pm

Thanks for sharing 😊

Like 0 | Share



Aveek Kumar Baruah

February 16, 2020 at 3:17 pm

I don't think the lines

```
convert date sy-datum time sy-uzeit  
into time stamp l_timestamp time zone l_timezone.
```

work as intended, as the value of l_timestamp is 0.

Like 0 | Share



Sandra Rossi

February 16, 2020 at 5:33 pm

I don't see why standard ABAP shouldn't work as [intended](#). Check SAP notes to find out what your problem is...

Like 0 | Share

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	
Newsletter	Support