Former Member

April 18, 2012 | 5 minute read

Create and view LOG using SLG0 and SLG1 transaction

Create and view LOG using SLG0 and SLG1 transaction

Application LOG

When we need to display all messages together then this set of messages is a log. Logs usually also contain general header information (log type, creator, creation time, etc.). Several logs can be created in a transaction. The Application Log provides a comprehensive infrastructure for collecting messages, saving them in the database and displaying them as logs. It has also traffic icons for the messages based on the message type.

Important Transactions

The important transactions codes used for application log are as following:

- **SLG0** – Create a new Log Object and sub object
- **SLG1** – Display Application Log
- **SLG2** – Delete the Application Log

Creation of object and sub object:

For generating a custom application log we need to create a new log object and sub object. For this we use transaction SLG0.

Go to transaction SLG0 and click on new entries. Then give the name of object and save. For any of the object we can create the sub object as well. Sub objects are simply further classifications of the application log.

To collect messages and display them in LOG in program:

Function modules to be used:

BAL_LOG_CREATE Create log with header data

BAL_LOG_MSG_ADD Add a message to a log

BAL_DB_SAVE Save the messages

BAL_DSP_LOG_DISPLAY Display message in memory

Open Log

The FM BAL_LOG_CREATE is used to open application log. This FM returns log handle. This log handle is a unique identifier for a particular log. We use this log handle later for accessing the log like for example adding messages to the log etc.

Add message to LOG

Using the log handle we can add as many messages we want to the LOG through the FM BAL_LOG_MSG_ADD to the importing parameter I_S_MSG (structure BAL_S_MSG). This data is very much similar to the T100 data that is message type, message number and the four message variables.

Save messages

Logs there in the memory can be saved to database using the FM BAL_DB_SAVE by passing the importing parameter I_SAVE_ALL = 'X'.The function module BAL_DB_SAVE returns a table (Exporting parameter E_NEW_LOGNUMBERS), which relates LOG_HANDLE, external number EXTNUMBER, temporary LOGNUMBER and permanent LOGNUMBER, so you can find out which number was assigned to a log after saving.

Display messages

FM BAL_DSP_LOG_DISPLAY is used to display collected messages.

Below is one sample code using the above mentioned FMs:

```
ls_log-object = lc_object.          "Object name

 ls_log-aluser = sy-uname.          "Username

 ls_log-alprog = sy-repid.          "Report name

***Open Log

 CALL FUNCTION 'BAL_LOG_CREATE'

   EXPORTING

    i_s_log                 = ls_log

   IMPORTING

    e_log_handle            = ls_log_handle

   EXCEPTIONS

    log_header_inconsistent = 1

    OTHERS                  = 2.

 IF sy-subrc EQ 0.
```

```
***Create message
   ls_msg-msgty = lc_type.          "Message type
   ls_msg-msgid = lc_msgid.         "Message id
   ls_msg-msgno = lc_msgno.         "Message number
   ls_msg-msgv1 = lv_message1.      "Text that you want to pass as message
   ls_msg-msgv2 = lv_message2.
   ls_msg-msgv3 = lv_message3.
   ls_msg-msgv4 = lv_message4.
   ls_msg-probclass = 2.
   CALL FUNCTION 'BAL_LOG_MSG_ADD'
    EXPORTING
     i_log_handle            = ls_log_handle
     i_s_msg                 = ls_msg
* IMPORTING
*   E_S_MSG_HANDLE          =
*   E_MSG_WAS_LOGGED        =
*   E_MSG_WAS_DISPLAYED     =
   EXCEPTIONS
```

```
        log_not_found           = 1
        msg_inconsistent        = 2
        log_is_full             = 3
        OTHERS                  = 4.
    IF sy-subrc NE 0.
     "Do nothing
    ENDIF.
    INSERT ls_log_handle INTO TABLE li_log_handle.
***Save message
    CALL FUNCTION 'BAL_DB_SAVE'
     EXPORTING
      i_client             = sy-mandt
*   I_IN_UPDATE_TASK      = ' '
      i_save_all           = lc_set
      i_t_log_handle       = li_log_handle
* IMPORTING
*   E_NEW_LOGNUMBERS      =
     EXCEPTIONS
```

```
log_not_found        = 1
save_not_allowed     = 2
numbering_error      = 3
OTHERS               = 4.
IF sy-subrc EQ 0.
 REFRESH: li_log_handle.
ENDIF.
```

Usage of Application LOG

We can use application log in the below scenarios:

- When we want to save the List/Log for the long time: Generally, we have the spool retention period of 8 days. So, the list or log will be deleted automatically.
- When we want more information compared to Log generated with WRITE: Application Log has more information like User, date of creation, Severity of the message.
- In ALE / EDI Processing: When we do the cross client processing (IDoc Processing), we can generate the Application log to keep track of the generated errors or the messages.

Summary of the steps to Create and View logs:

1) 1) Create object and sub object in SLG0 transaction.
2) 2) The program where you want to create the LOG call the FMs:

```
BAL_LOG_CREATE

BAL_LOG_MSG_ADD
BAL_DB_SAVE
```

3) 3) For viewing the logs go to SLG1 transaction and give the object name, sub object name (if any) and other related information like the Username and date etc.

4) Then click on Execute. You will be able to see the logs. Double click on any one of them to see the detailed error message.

Follow

4) One can even view the logs through the program itself by using the FM BAL_DSP_LOG_DISPLAY.

Alert Moderator

Like

Assigned Tags

ABAP Development

abap

Similar Blog Posts

Create Error Logging in your applications using SLG0, SLG1

By Philip Johnston Feb 27, 2013

Global Class and Method for Application Log

By Debopriya Ghosh May 08, 2014

How to display additional information about IDoc processing in the Application Log - Developer Guideline

By Andrea Olivieri Jan 14, 2013

Related Questions

application log

By Hymavathi Oruganti Mar 09, 2012

need transaction or table

By Former Member Feb 26, 2007

Refresh Button in SLG1

By Aaron Kitts Dec 07, 2017

Follow

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.

12 Comments

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

Former Member

September 5, 2013 at 3:14 pm

Good Job !!!

Like 0 | Share

Follow

Former Member
September 5, 2013 at 3:32 pm

Good One 🙂

Like 0 | Share

Former Member
January 13, 2014 at 12:41 pm

Good article , very descriptive.

one question how many days by default we have our logs in SLG1?

Like 0 | Share

César Augusto Scheck
July 2, 2014 at 2:46 pm

Hi Akankshi,

Nice posting, it's helped me a lot.
This article cuts to the chase and you even posted a cool summary to speed up learning.

Do you know if is there some ABAP class that could provide/encapsulate the same functionality?

Best regards,

Like 0 | Share

Sanjay Shah
June 27, 2018 at 11:40 am

CL_BAL_LOGGER

Like 0 | Share

Former Member
September 3, 2014 at 6:43 am

Thank akankshi,

I found this article very helpful.

Regards,

Punleu

Like 0 | Share

Jatan Saraf
February 4, 2016 at 2:27 pm

Very helpful article....very well written!!

Like 0 | Share

Follow

Former Member

August 23, 2016 at 11:01 am

Hi Akankshi,

Currently in SLG1 we can see failures and warning messages which are in red/yellow light, may I know how we can display success logs which are in green light in SLG1?

Do we need to change FM or any configuration in system?

Regards

NTG

Like 0 | Share

Former Member

April 6, 2017 at 11:42 am

Very nice blog

Like 0 | Share

Former Member
May 10, 2017 at 6:39 pm

Very Good.

Still useful in 2017

Thanks

Like 0 | Share

Bhakti joshi
March 30, 2020 at 7:31 pm

i like the way you have laid out here. This format of detailed step wise explanation preceded and followed by summary makes it easy to grasp and helps simplify. thanks for the effort, totally worth it.

Like 0 | Share

Jigang Zhang 张吉刚
April 28, 2021 at 2:05 am

concise and very helpful about how to deal with application logs!

Like 0 | Share

Follow

Find us on

Like

RSS Feed

ivacy	Terms of Use
Legal Disclosure	Copyright
Trademark	
Newsletter	Support