



## SAP NEWS

# Create Application Log Using SLG0, SLG1 and SLG2

[Home](#) / [ABAP](#) / Create Application Log Using SLG0,...



## Manage From Cloud or On-prem

Streamline updates across your fleet and automatically save secure passwords.

[Learn](#)

**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**

Please note that this type of application log comes handy when we want to add a log to proxy interfaces. Message in this type of application log can be viewed any time.

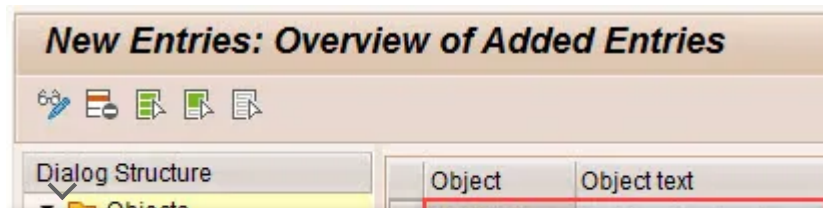
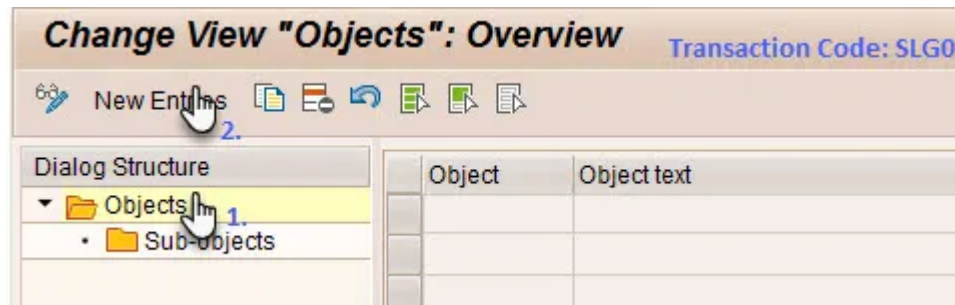
The following main transactions are used for application log:

- SLG0 : To create log 'Object' and 'Sub Object'
- SLG1 : To display the application log entries
- SLG2 : To delete the application log entries

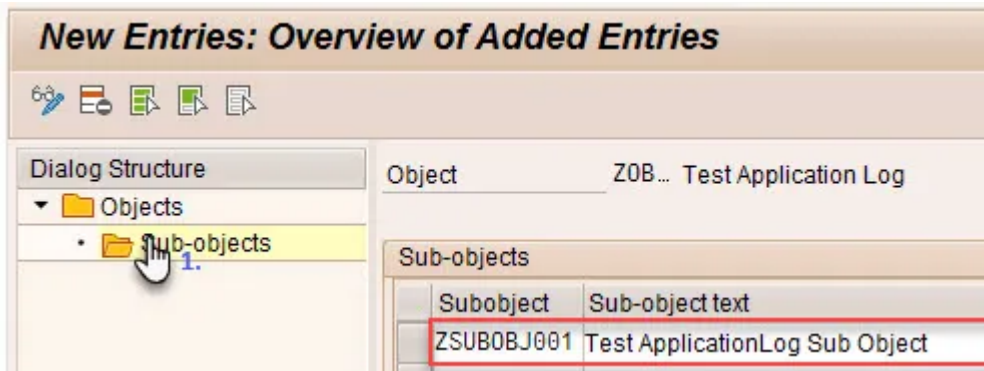
### **Step 1 – Create Application Using SLG0:**

The first step is to create the application log in transaction 'SLG0' before we can populate the log via an ABAP program. In 'SLG0' create an 'Object' together with its 'Sub Object' using the following steps:

- Create the object 'ZOBJ001'



**500+ Policies, Password Alert, &  
Sandboxing — keep your employees safe  
wherever they are.**



- Save

## Step 2 – Create ABAP program to populate the Application Log:

Please find below a sample program which demonstrate how to populate the application log created in step 1.

```

1  REPORT ztestapplog.
2  *&-----&*
3  *& Program Description: &*
4  *& -----&*
5  *& This demo program demonstrate how to create application log via &*
6  *& transaction code 'SLG0' and display the log via 'SLG1'. &*
7  *& &*
8  *& Author: ABAPCOOKBOOK &*
9  *& Website: www.abapcookbook.com &*
10 *****
11
12 *****
13 * DATA DECLARATIONS *
14 *****
15 *Constants:
16 ✓CONSTANTS:
17

```

**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**

```

26  gc_probclass_low          TYPE bal_s_msg-probclass  VALUE '4',
27  gc_problem_class_other    TYPE c                  VALUE ''.
28
29  *Variables:
30  DATA:
31  gv_log_handle TYPE balloghndl,
32  gv_msgv1      TYPE symsgv.
33
34
35  *****
36  * CREATE LOG. *
37  *****
38  *IMPORTANT: Make sure the object 'ZOBJ001' and sub-object 'ZOBJ001' has
39  *          been created in transaction 'SLG0'. Log can be deleted via
40  *          transaction 'SLG2'.
41
42  *          The log can be viewed via transaction 'SLG1' using
43  *          the object 'ZOBJ001' and sub-object 'ZOBJ001'.
44
45  *Create the application log.
46  PERFORM f_create_bal_log USING gv_log_handle.
47
48
49  *****
50  * ADD MESSAGE TO LOG. *
51  *****
52  *This is a test condition to add a message.
53  IF 1 NE 2.
54
55  * Typecasting the date.
56  gv_msgv1 = sy-datum.
57
58  * Add a message to the application log.
59  * Message: 'Date & is not valid'
60  PERFORM f_add_msg_to_log USING gv_log_handle          " Log Handler
61                                gc_problem_class_important " Class Problem Level
62                                gc_error                 " Error Message
63                                gc_message_id            " Message Class
64                                gc_msgno1               " Message Number
65                                gv_msgv1                " Message Dynamic Content(&) 1
66                                space                   " Message Dynamic Content(&) 2 (Fmptv)

```

**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**

```

75 *****
76 *At the end of all operation, save the application log.
77 *The log can be viewed via transaction 'SLG1' using
78 *the object 'ZOBJ001' and sub-object 'ZOBJ001'.
79 PERFORM f_save_bal_log USING gv_log_handle.
80
81
82 *****
83 * ROUTINES: *
84 *****
85
86 *-----*
87 * Sub routine to create the application log for error logging *
88 * purposes. *
89 *-----*
90 FORM f_create_bal_log USING pe_log_handle TYPE balloghndl.
91 *-----*
92 * Local data declarations.
93 * Structures.
94 DATA:
95     lst_log          TYPE bal_s_log.
96
97 * Defining some header data of the application log.
98 lst_log-aldate      = sy-datum.
99 lst_log-altime      = sy-uzeit.
100 lst_log-aluser      = sy-uname.
101 lst_log-alprog      = sy-repid.
102 lst_log-object      = gc-object.
103 lst_log-subobject   = gc-subobject.
104
105 * Creating the application log.
106 CALL FUNCTION 'BAL_LOG_CREATE'
107     EXPORTING
108         i_s_log      = lst_log
109     IMPORTING
110         e_log_handle = pe_log_handle
111     EXCEPTIONS
112         OTHERS      = 1.
113 * Not necessary to cater for this exception.
114
115 ✓ FNDFORM.

```

**500+ Policies, Password Alert, &  
Sandboxing — keep your employees safe  
wherever they are.**

```

124         pi_msgid TYPE bal_s_msg-msgid
125         pi_msgno TYPE bal_s_msg-msgno
126         pi_msgv1 TYPE bal_s_msg-msgv1
127         pi_msgv2 TYPE bal_s_msg-msgv2
128         pi_msgv3 TYPE bal_s_msg-msgv3
129         pi_msgv4 TYPE bal_s_msg-msgv4
130         pi_detlevel TYPE bal_s_msg-detlevel.
131 *-----*
132 * Structures:
133 DATA:
134     lst_ballog_msg          TYPE bal_s_msg.
135
136 * Defining data of message for the application log.
137 lst_ballog_msg-probclass = pi_probclass.
138 lst_ballog_msg-msgty    = pi_msgty.
139 lst_ballog_msg-msgid     = pi_msgid.
140 lst_ballog_msg-msgno     = pi_msgno.
141 lst_ballog_msg-msgv1     = pi_msgv1.
142 lst_ballog_msg-msgv2     = pi_msgv2.
143 lst_ballog_msg-msgv3     = pi_msgv3.
144 lst_ballog_msg-msgv4     = pi_msgv4.
145 lst_ballog_msg-detlevel  = pi_detlevel.
146
147 * Adding this message to log file.
148 CALL FUNCTION 'BAL_LOG_MSG_ADD'
149     EXPORTING
150         i_log_handle = pi_log_handle
151         i_s_msg      = lst_ballog_msg
152     EXCEPTIONS
153         log_not_found = 0
154         OTHERS        = 1.
155 * Not necessary to cater for these exceptions.
156
157 ENDFORM.
158
159
160 *-----*
161 * Sub routine to save the application log to database.
162 *-----*
163 FORM f_save_bal_log USING pi_log_handle TYPE balloghndl.
164 *-----*

```

**500+ Policies, Password Alert, &  
Sandboxing — keep your employees safe  
wherever they are.**

```
173 CALL FUNCTION 'BAL_DB_SAVE'
174     EXPORTING
175         i_t_log_handle    = lt_loghandle
176     EXCEPTIONS
177         log_not_found     = 1
178         save_not_allowed  = 2
179         numbering_error   = 3
180         error_message     = 4
181         OTHERS            = 5.
182 * Not necessary to cater for these exceptions.
183
184 ENDFORM.                                "F_SAVE_BAL_LOG
```

### **Step 3 – Displaying application log in SLG1:**

Execute the above program, an entry will be added to the application log. This log entry can be viewed via transaction 'SLG1'.

In transaction, please enter the object 'ZOBJ001' and sub object 'ZSUBOBJ001' (created in step 1) and execute.



**500+ Policies, Password Alert, &  
Sandboxing — keep your employees safe  
wherever they are.**

Analyze Application LogTransaction Code: SLG1

Object

ZOBJ001

Test Application Log

Subobject

ZSUBOBJ001

(generic input)

External ID

\*

Time Restriction

From (Date/Time)

29.05.2017

00:00:00

To (Date/Time)

29.05.2017

23:59:59

Log Triggered By

User

\*

Transaction code

\*

Program

\*

Log Class

☐ Only very important logs

☐ Only important logs

☐ Also less important logs

☒ All logs

Log Creation

☒ Any

☐ Dialog

☐ In batch mode

☐ Batch input

Log Source and Formatting

☒ Format Completely from Database

☐ Format Only Header Data from Database

☐ Format Completely from Archive

**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**



**Display logs**

Technical Information
 Help

Date/Time/User	Nu...	Externa...	Object text	Sub-object text	Transact...	Program	Mode	Log number
▼ 29.05.2017 17:29:13	1		Test Application...	Test Application...	SE38	ZTESTAPPLOG	Dialog pro...	0000000000...
• 29.05.2017 17:29:13	1		Problem class Imp					
▶ 29.05.2017 17:29:15	1		Test Application...	Test Application...	SE38	ZTESTAPPLOG	Dialog pro...	0000000000...
▶ 29.05.2017 17:29:20	1		Test Application...	Test Application...	SE38	ZTESTAPPLOG	Dialog pro...	0000000000...

0
 1
 0
 0

Ty... Message Text
 

Date 20170529 is not valid

#### Step 4 – Delete log entries in SLG2:

Finally, log entries can be deleted via transaction code ‘SLG2’.

**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**

## Application Log: Delete Expired Logs

Transaction Code: SLG2



### Delete logs

All logs are deleted which satisfy the following selection conditions, and for which:

- the expiry date is reached or passed
- the expiry date is not defined

### Expiry date

- ☐ Only logs which have reached their expiry date
- ☐ and logs which can be deleted before the expiry date
- ☒ Cannot delete log now since expiry date is in the future

### Selection conditions



Object	ZOBJ001	to		
Subobject	ZSUBOBJ001	to		
External ID		to		
Transaction code		to		
User		to		
Log number		to		
Problem class		to		
from (date/time)			00:00:00	
to (date/time)			00:00:00	

### Options

- ☒ Only calculate how many
- ☐ Generate list
- ☒ Delete immediately

**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**

Voila 😊 hope this helps. You can also refer to this [tutorial](#) on how to display application log automatically in a program.



## Remote U.S Software Jobs

U.S Software Developers Jobs. Work Remotely. No Vis  
Full time jobs. High salary.

Turing.com

Categories: ABAP, Tutorial By ABAPCOOKBOOK EDITOR May 29, 2017

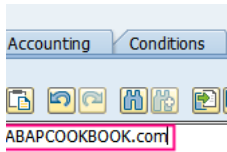
Tags: [abap](#) [application log](#) [code](#) [create log](#) [display application log in SLG1](#) [log](#) [sample program](#) [sap](#) [SLG0](#) [SLG1](#) [SLG2](#) [Tutorial](#)

Share this post



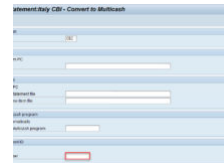
500+ Policies, Password Alert, &  
Sandboxing — keep your employees safe  
wherever they are.

## Related posts



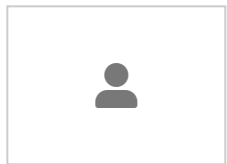
How To Use Function Module READ\_TEXT

September 23, 2018



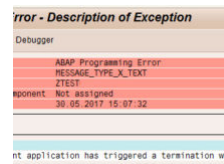
Solve Error 'File Has Already Been Loaded' in FEBC

June 22, 2017



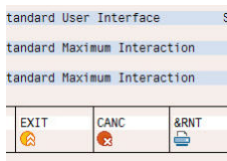
How To Quickly Save ABAP Code To Your Computer

June 2, 2017



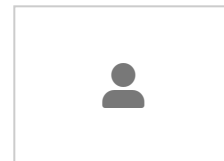
SAP ABAP Message Types

May 30, 2017



Back Button Not Working in GUI-STATUS/PF-STATUS

May 29, 2017



Change Variant Content Dynamically

March 13, 2017

Type and hit enter ...



**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**

ABAP

ABAP Performance

BAPI

Code Sample

For Dummies

Interface

SAP Configurations

SAP Enhancement

SAP News

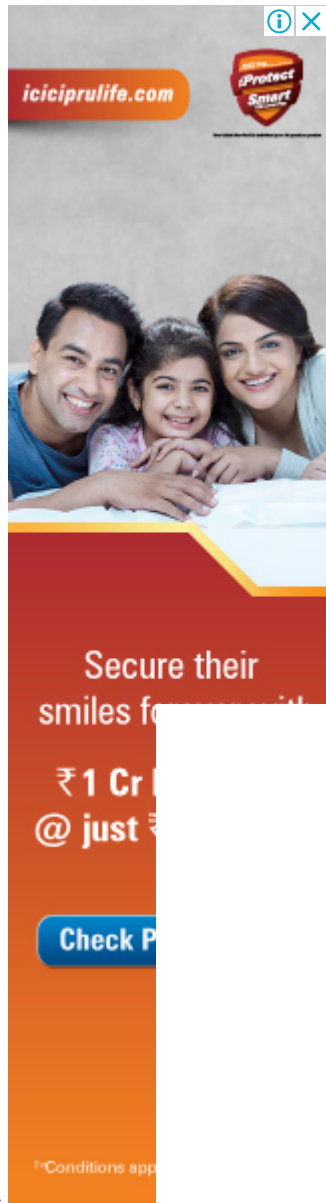
Smartforms

Tips & Tricks

Tutorial



**500+ Policies, Password Alert, &  
Sandboxing — keep your employees safe  
wherever they are.**



**500+ Policies, Password Alert, & Sandboxing — keep your employees safe wherever they are.**



**500+ Policies, Password Alert, &  
Sandboxing — keep your employees safe  
wherever they are.**