

# User Exit 12 (Multiple Dataset Log Processing)

**Note:**

UEX2 and UEX12 are mutually exclusive for an Adabas nucleus session: only one can be specified.

This user exit is given control by the Adabas nucleus during a switch from one multiple log dataset to another for the purpose of copying the log dataset before it is reused by Adabas. This switch occurs only if multiple dataset data protection logging and/or multiple dataset command logging is in effect for the session.

The user exit routine is designed to invoke a procedure that will execute the appropriate function (CLCOPY or PLCOPY) of the ADARES utility.

User exit 12 is invoked

- during Adabas nucleus startup if a multiple PLOG/CLOG dataset has to be copied;
- whenever a switch to another log dataset occurs;
- during Adabas nucleus shutdown.

The user exit is provided with information about the type of log (PLOG or CLOG) and the status of the multiple log datasets.

The user exit can decide which action is to be taken:

- Ignore the call and allow Adabas to proceed;
- Submit a job to copy and mark as empty the log dataset just filled (ADARES utility);
- Direct Adabas to wait for a specified interval, then call the user exit again with updated PLOG/CLOG dataset status information. During the wait interval, no commands that may produce log records for the log type being processed are allowed to proceed.

An example of user exit 12 is supplied with the Adabas installation procedure. Refer to the *Adabas Installation* documentation for more information.

The call to the user exit is made using a standard BASSM R14,R15 Assembler instruction. All registers must be saved when control is received and restored immediately prior to returning control to Adabas. Register 15 contains an action code as described on page 19.

**Note:**

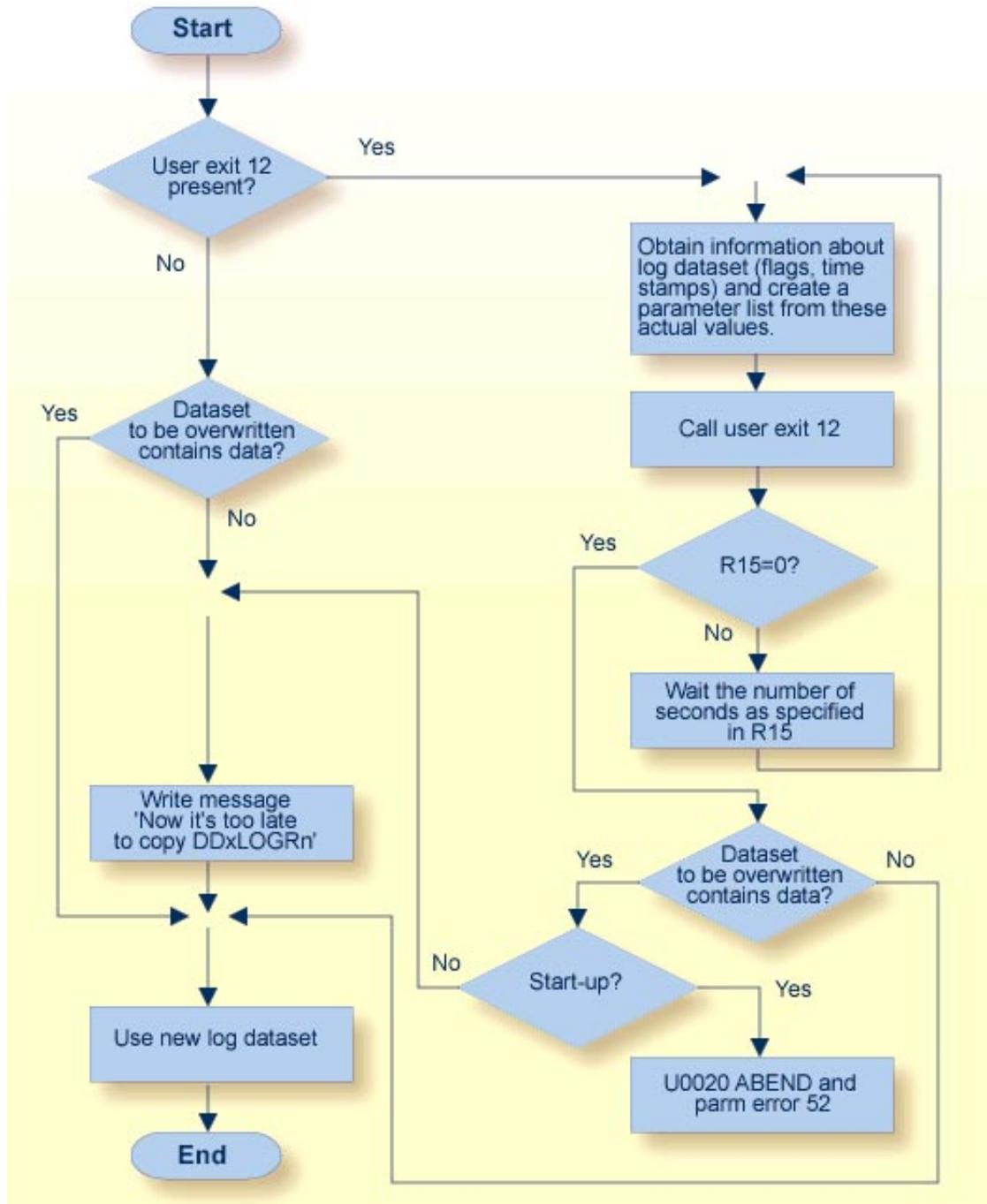
User exit 12 must return the same AMODE value to the calling program that was active when user exit 12 was called. The recommended Assembler instruction to return is BSM 0,R14.

This chapter covers the following topics:

- User Exit 12 Calling Sequence

- User Exit Interface
- Output Parameter
- Activating the Sample User Exit

## User Exit 12 Calling Sequence



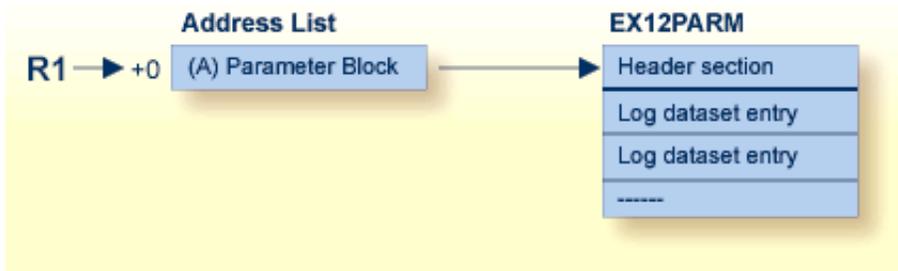
## Multiple Log Processing Flow

# User Exit Interface

Optionally, the user exit may initialize its operation. It may store any value in field EX12USER of the EX12PARAM parameter block to keep track of its resources. This field is considered as "owned" by the user exit and is supplied again for all subsequent executions of the exit. It is set to zero when the exit is first called and is not modified by Adabas thereafter.

The user exit is called again during termination to do any necessary finishing or 'cleanup' work.

## Parameters



## User Exit 12 Parameters

### DSECT of the EX12PARAM Parameter Block

```

Macro
EX12PARAM
*****
.* Name           Exl2Parm                      *
.* Component      ADABAS User Exit              *
.* Function       Parameter list for User Exit 12 *
.*               (replacement for User exit 2 for use when there are *
.*               more than 2 PLOGs or CLOGs)    *
.* Parameters     None                          *
.* Restrictions   None                          *
.* Notes         None                          *
*****

.*
EX12PARAM DSECT ,           User Exit 12 Parameter List
*
EX12HDR  DS      0F          Common header section
*
EX12USER DS      F          Reserved for the user. This field +
                           is initialized to zero before the +
                           exit is called the first time, and +
                           will not be altered by ADABAS after +
                           that. It can be used to maintain +
                           information across invocations.

*
EX12LOGT DS      X          Log type
EX12PLOG EQU     C'P'      PLOG
EX12CLOG EQU     C'C'      CLOG
*
    
```

```

EX12TYPE DS      X          Call type
EX12TBEG EQU     C'S'      Nucleus start
EX12TSW EQU     C'W'      Log switch
EX12TEND EQU     C'T'      Nucleus termination
                DS      XL2      Reserved
*
EX12NLOG DS      F          Number of logs
EX12DBID DS      F          Database ID
EX12NUCI DS      F          Nucleus ID
*EX12PLGN DS     F          Current session PLOG number      +
                        (zero for CLOGs)
EX12NCMP DS      F          Log just completed
EX12STAT DS      X          Flags of next log in sequence
EX12WNUC EQU     X'80'     Being written by the nucleus
EX12FULL EQU     X'40'     Completed by the nucleus
EX12RES EQU     X'20'     Being copied by ADARES
EX12CL5 EQU     X'08'     CLOGLAYOUT=5
EX12UNUS EQU     X'00'     Unused and/or copied
                DS      XL3      Reserved
                DS      4F      Reserved
*
EX12HDRL EQU     *-EX12HDR  Length of header section
*
EX12LOG DS       0F          Start of individual log dataset      +
                        entries. This section is repeated      +
                        for the number of logs specified in      +
                        field EX12NLOG
*
EX12LTIM DS      XL8       Time stamp of write to log dataset
EX12LNUM DS      F          Number of log dataset
EX12LFLG DS      X          Flags (mapped as in EX12STAT)
                DS      XL3      Reserved
                DS      4F      Reserved
*
EX12LOGL EQU     *-EX12LOG  Length of a log dataset entry
*
*
                MEnd

```

## Output Parameter

Parameter	Usage
R15 = 0	Nucleus continues processing.
R15 > 0	R15 is treated as the number of seconds to wait before calling user exit 12 again with updated status for all log datasets. During this time, no commands that may create log entries are processed.

## Activating the Sample User Exit

The sample user exit is written in Assembler language. It performs the following functions:

- Issues a message identifying the reason and the type of log for which it was called.

- Issues a message with the status and timestamp of all log datasets that are not empty.
- If any log dataset is full and at least one log dataset has a status that is different from the last time the exit was called, the exit reads 80-byte records from an input file and writes them to an output file. It replaces all occurrences of the character "?" with either "P" or "C", depending on whether the exit was invoked to process a PLOG or CLOG event, respectively. This allows the input file to accommodate an event for either log type. Normally, the input file contains job control statements and the output file is directed to a job execution queue.
- If at least one log dataset is not full, the exit returns to the caller with R15 zero, which allows Adabas to proceed.
- If all log datasets are full, the exit returns to the caller with R15 nonzero, which directs Adabas to wait for the number of seconds in R15, then call the exit again with an updated status of all log datasets. The default delay time is 30 seconds.

## OS/390 or z/OS

The sample user exit UX12SAMP is delivered on OS/390 and z/OS as source and as a load module that can be used without change or reassembly.

The source and load forms of the user exit are delivered in the Adabas source and load libraries, respectively. The job to assemble the user exit UX12ASML is located in the Adabas jobs library. The jobs library also contains a sample job UX12CJOB to be customized and submitted by the user exit that invokes the ADARES utility PLCOPY or CLCOPY function.

Activate the sample user exit as follows:

1. In addition to ADARUN NxLOG={2-8}, specify ADARUN UEX12=UX12SAMP for the Adabas nucleus.
2. Supply the job control model that the user exit is supposed to submit under the DDNAME COPYJOB.

Provide the following DD statement:

```
//INTRDR2 DD SYSOUT=(*,INTRDR)
```

### Note:

The same DD statement is used by the sample user exit 2 or 12 for submitting PLCOPY or CLCOPY jobs.

## VSE/ESA

The sample user exit UX12SAMP is delivered on VSE/ESA as source and as a phase that can be used without change or reassembly.

The source A.book and phase are delivered in the Adabas VSE sublibrary. The job to assemble the user exit UX12ASML.X is also provided in the Adabas sublibrary. The sample job UX12CJOB.X is to be customized and submitted by the user exit that invokes the ADARES utility PLCOPY or CLCOPY function.

Activate the sample user exit as follows:

1. In addition to ADARUN NxLOG={2-8}, specify ADARUN UEX12=UX12SAMP for the Adabas nucleus.
2. Supply the job control model that the user exit is supposed to submit under the DLBL COPYJOB.

## BS2000

The sample user exit UX12SAMP is delivered on BS2000 as source and as an object module that can be used without change or reassembly.

The source form of the user exit is delivered in the Adabas source library. The job to assemble the user exit, UX12ASML, is located in the Adabas jobs library. The jobs library also contains a sample job UX12CJOB to be submitted by UX12SAMP that invokes the ADARES utility PLCOPY or CLCOPY function.

Activate the sample user exit as follows:

1. In addition to ADARUN NxLOG={2-8}, specify ADARUN UEX12=UX12SAMP for the Adabas nucleus.
2. If the machine can process job variables, place the following statement in the Adabas nucleus job:

```
/SET-JV-LINK LINK-NAME=*DDJBDSF , -
/          JV-NAME=#JV.DSF
/MODIFY-JV   JV-ID=JV-NAME(JV-NAME=#JV.DSF) , -
/          VALUE='ENTER-JOB FROM-FILE=*LIB-ELEM( , -
/LIB=library,ELEM=dsfjob) , -
/JOB-CLASS=JCDSF'
```

where

*library* is the library; and  
*dsfjob* is the member where the customized job from step 2 is stored.

If the machine cannot process job variables, Adabas attempts to submit the job

```
(ADABAS.SRC,UX12CJBP,J)
```

for the PLCOPY job and

```
(ADABAS.SRC,UX12CJBC,J)
```

for the CLCOPY job.

An example job that allows the database ID to be supplied as a parameter is supplied in the Adabas source library ADAvrs.SRC.

If this is not satisfactory, you can modify the supplied source UX12SAMP.

When using user exit 12 with BS2000 systems, the name of the ADARES CLCOPY or PLCOPY job is no longer "hard-coded" in the user exit as in previous releases. Therefore, there are now two ways of specifying the /ENTER job:

## Method 1

Create a job variable containing the complete "enter job" command. For example:

```
/DCLJV  ENTER.ADARES.PLCOPY, LINK=*DDJBPLC
/SETJV  *DDJBx12,C'ENTER-PROC(srclib,ux12cjbx),(DB=dbid),J-N=X12COPY'
```

where

*srclib* is the library where the UX12CJBx(J) is stored (x=P for the PLCOPY job and x=C for the CLCOPY job)

*ux12cjbx* is the J element name of the respective procedure job.

*dbid* is the database ID of this nucleus.

Note that all operands of the /ENTER command may be used. The presence of the /DCLJV statement overrides the other possibilities. If necessary, the content of the job variables can be changed during a nucleus session.

Use the link names "\*DDJBPLC" for assigning the ADARES PLCOPY job, and "\*DDJBCLC" for assigning the ADARES CLCOPY job.

## Method 2

Omit the JV specification. The user exit 12 will then issue the following command:

```
'ENTER-PROC RES.EPdbid.xLCO,(DB=dbid),J-N=xL12<dbid>'
```

Use the following code to extract the job element in your database job deck prior to starting the database:

```
/DELETE-FILE RES.EPdbid.CLCO
/SET-JOB-STEP
/DELETE-FILE RES.EPdbid.PLCO
/SET-JOB-STEP
/ASS-SYSDTA *SYSCMD
/STA-LMS
//O srclib
//EXT-E (,ux12cjbc,J),RES.EPdbid.CLCO
//EXT-E (,ux12cjbpc,J),RES.EPdbid.PLCO
//END
/ASS-SYSDTA *P
```

where

*dbid* is the database ID of the P/CLCOPY database (this could be &DBID. in a procedure)

*srclib* is the source library where the template copy procedures are held

*ux12cjbc* is a procedure to perform the exit 12 ADARES CLCOPY job

*ux12cjbpc* is a procedure to perform the exit 12 ADARES PLCOPY job

**Note:**

For BS2000 systems, user exit 12 is delivered both as a source and as an object element.