SDSFEXT User's Guide

Notification of Batch Job Abnormal Completion

By Lionel B. Dyck
Kaiser Permanente Information Technology
25 N. Via Monte Ave
Walnut Creek, California 94598
e-mail: lionel.b.dyck@kp.org

This document defines a new facility that is available for use in both production and adhoc batch jobs to provide electronic mail notification to responsible individual(s) about abnormal completion status.

Contents

Overview

When a JOB is submitted for batch execution the submitter or responsible programmer must wait for the JOB to complete execution and for the output to be printed and reviewed before they are informed of any anomalies in the execution of the JOB. If they are really anxious to know the completion status they can LOGON to TSO and use SDSF, or similar product, to review the execution status. This is acceptable practice assuming the JOB completes execution during normal working hours and the individual has the time to review the JOBs execution status.

A more effective use of the responsible individual(s) time is to have add a step to the batch JOB that will generate an electronic mail notification only when the JOB ends with an abnormal completion code.

Details

The SDSFEXT, short for SDSF Extract, is a utility designed to be executed as the last step of a batch JOB which can extract from the JES2 SPOOL selected SYSOUT data sets. The extracted data can then be sent via E-Mail to the responsible individual(s) who can then review the cause of any anomalies in the execution of the JOB.

A simple to use PROC has been installed under the name SDSFEXT (see Figure 1) which can be used to invoke the SDSFEXT utility.

As an example of using SDSFEXT is documented in Figure 2 below. This example executes the program IEHLIST in stepname HLIST.

For this example:

- the processing of the IEHLIST will fail with a return code of 8 (because there is no DD allocation for the requested volser being listed).
- The JCL IF/THEN/ENDIF logic is used to test the return code from the stepname HLIST being greater than 0. If so then the JCL between the IF and ENDIF will be processed (this is much cleaner syntax than using the older JCL conditional (COND) keyword).
- The SDSFEXT Proc is executed letting the JOBNAME default to * (see Figure 1) and specifying that the three system DDnames and the SYSPRINT from step HLIST be extracted to the temporary output data set defined by the OUTDD keyword.
- The OUT DD statement must be coded immediately after the SDSFEXT Proc statement and defines a temporary data set that will be passed to the next step. This data set will contain the data from the requested DDnames by SDSFEXT.
- The TSOB step executes the Batch TSO program which executes the TSO Command XMITIP to e-mail the data in the passed data set to the responsible individual for the job. The CC defines a second responsible individual. See the XMITIP documentation or use the XMITIP ISPF dialog for further information.

The sample JCL in Figure 3 can be copied into any Production or adhoc Batch JOB with minor tailoring for use.

Some things to be aware of:

1. The JCL IF syntax can test for a stepname or a stepname.procstepname.
2. The condition code testing is very extensive – see the JCL User's Guide for full details.
3. The examples demonstrate testing for a condition code greater than 0. In some cases a return code of 4 may be acceptable for a JOB.
4. Extracting and e-mailing large reports can cloud the issue of problem identification and resolution so only include those DDnames that will prove helpful. Remember that the JOB will be available in SDSF or CA-View if more information is necessary.
5. Use this for quick notifications and not as a single source for capturing information on the failure of a JOB.

SDSFEXT Command Syntax

The syntax for the SDSFEXT command is:

SDSFEXT jobname ddname(s) jobnumber (options

jobname	Defines the name of the JOB to select. The name may also be specified as * if processing the currently active JOB.
ddname(s)	Defines the DDname within the JOB to select. The DDname may be coded as: ddname stepname.ddname stepname.procstepname.ddname (ddname1 ddname2 …)
jobnumber	**Optional** and not used if processing the currently active JOB. Useful if processing a JOB in the JES2 SPOOL for which there are multiple occurrences of the same JOB name to select the specific JOB of interest.
(	If any options are specified than the (is required.
options	Valid options are: **ACTIVE** – used to indicate that the specified JOB is currently in execution. **DEBUG** – used to cause the command to echo to the SYSTSPRT DD information about its processing. This will only be useful to the systems programmer responsible for supporting SDSFEXT. **INPUT** – used to indicate that you will be requesting to extract input data (e.g. SYSIN).

Examples

```
//SDSFEXT  PROC DEBUG=,JOBNAME='*',ACTIVE=
//* -------------------------------------------------------- *
//* SDSF EXTRACT PROC TO EXTRACT A SPECIFIC DDNAME FROM THE  *
//* JES2 SPOOL TO A SPECIFIED OUTPUT DD FOR THE ACTIVE JOB.  *
//*                                                          *
//* SYMBOLICS ARE:                                           *
//*    ACTIVE  IS ACTIVE IF USING JOBNAME OF *               *
//*    DDNAME  IS THE DDNAME TO SELECT                       *
//*    DEBUG   IS TO ENABLE DIAGNOSTIC INFORMATION           *
//*    INPUT   ENABLES ACCESS TO INPUT DATA                  *
//*    JOBNAME IS THE NAME OF THE JOB TO SELECT OR * FOR     *
//*            THE CURRENT (ACTIVE JOBNAME)                  *
//*    OUTDD   IS THE OUTPUT DD DD THAT THE USER PROVIDES    *
//* -------------------------------------------------------- *
//TSOB    EXEC PGM=IKJEFT1B,
//  PARM='%SDSFEXT &JOBNAME &DDNAME &OUTDD (&ACTIVE &DEBUG
&INPUT'//SYSEXEC  DD  DISP=SHR,DSN=SYSLBD.LIONEL.EXEC
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  DUMMY
//         PEND
```

Figure 1: Sample SDSFEXT Procedure

```
//jobname  JOB  account,,NOTIFY=&SYSUID,
//        MSGLEVEL=(1,1)
//HLIST   EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 LISTVTOC VOL=3390=xxxxxx
/*
//TEST1 IF (HLIST.RC > 0 ) THEN
//SDSFEXT1 EXEC SDSFEXT,
//             DDNAME='(JESJCL JESMSGLG JESYSMSG HLIST.SYSPRINT)',
//             OUTDD=OUT
//OUT      DD  DISP=(MOD,PASS),UNIT=SYSDA,SPACE=(TRK,(15,15)),
//          DSN=&&OUT
//***************************************
//TSOB     EXEC PGM=IKJEFT1B
//SYSEXEC  DD  DISP=SHR,DSN=local.rexx.library
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
%XMITIP to-e-mail@address SUBJECT "JOB Report Failure" +
     CC second-e-mail@address +
     FILEDD OUT +
     FORMAT RTF/LAND NOMSG +
     FROM your-e-mail@address

//OUT      DD  DISP=(OLD,DELETE),DSN=&&OUT
//         ENDIF
```

Figure 2: Sample Batch Job Showing More Complete Example

```
..
//TEST1 IF (stepname.RC > 0 ) THEN
//SDSFEXT1 EXEC SDSFEXT,
//             DDNAME='(JESJCL JESMSGLG JESYSMSG stepname.SYSPRINT)',
//             OUTDD=OUT
//OUT      DD  DISP=(MOD,PASS),UNIT=SYSDA,SPACE=(TRK,(15,15)),
//             DSN=&&OUT
//*****************************************
//TSOB     EXEC PGM=IKJEFT1B,
//SYSEXEC  DD  DISP=SHR,DSN=YOUR.EXEC.LIBRARY
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
%XMITIP YOUR.MAIL@ADDRESS +
   SUBJECT 'SAMPLE SDSFEXT REPORT' +
   FILEDD OUT NOMSG +
   FROM YOUR.MAIL@ADDRESS +
   FORMAT RTF/LAND
//OUT      DD  DISP=(OLD,DELETE),DSN=&&OUT
//         ENDIF
```

Figure 3: Sample JCL Statements to Insert in Batch Job