Preparing Document Files During a Disaster

Overview

If a company has the misfortune to suffer a disaster that causes the Print and Mail Operations Center to become unavailable, measures need to be taken to print the documents at a secondary location. This document will describe a process to create a format of mission critical applications so they can be transferred to the new location for printing. Whereas there are many ways to support printing during a crisis, most methods require that resource lists and files are kept current. This document describes a method using **A**dvanced Function Printing **C**onversion and **I**ndexing **F**acility (ACIF) to bundle the necessary resources with the document file.

Traditional Method

Normally during a crisis, the designated "Hot Site" will have a remote printer definition where the output data will be routed using NJE (spool to spool) or another transmission method to have the data arrive at the site designated as the Print and Mail recovery site. Using this method requires the remote print site to have access to the AFP resources and knowledge of the paper stock needed during printing. If the "Hot Site" has printers defined on a channel extender connection, that may help resolve the resource situation. Since some sites have printers driven by servers, they require the data file to be routed to the server and AFP resources must be available at this server.

Method using ACIF

Our testing has shown that using ACIF eliminates the dependency on external resources because the necessary resources are embedded in the data file during the ACIF preparation step. A second benefit results from using current data extracted from the archival/bundling system or EOS in our installation. During the extract process using EOS, there is a need to assign AFP resource values to the print file. Assignment of the page definition, form definition, TRC usage, fonts, and paper stock required will occur during this operation.

Our standard processing steps will create an output document file, load this data into EOS, and then does an extract of this data directly to the print queue. During the extract step a referback statement will be used to assign AFP values to the document queued for printing. During a disaster the process will change slightly with the additional of the ACIF function. The document data will be loaded into EOS where the new process will extract the data to a temporary dataset. Other steps will identify the AFP resources and assign these values as parameters for the ACIF step. The ACIF step will process the extracted data using the AFP values to create a full MO:DCA formatted file where the necessary AFP resources are embedded at the beginning. This file will be printed at the remote printer using a DEST parameter of the OUTPUT card. The remote printer can be a JES queue or a server.

ACIF Building Steps

This new process is dependent on the following setup functions:

- A routine will be coded in REXX (DRPRINT) to process various tables and build the JCL needed to invoke ACIF and route the print file remotely. This JCL will automatically be submitted to the Internal Reader without any interaction.
- A form table must exist as a repository of information describing each EOS form. This table will identify the referback that is used for each document file needing to be printed. It must also include the spool and user where the data is stored since a site could have multiple spools. See the example below:

```
AY2E,P,BIN 253 ,L,OPAPYCOM
AY2G,P,BIN 187 ,L,OPAPY88D
  |  |    |       |    |
  |  |    |       |    |_____ (#5)Referback name
  |  |    |       |_____ (#4)Data type, L=line mode
  |  |    |_____ (#3)User Name for data
  |  |_____ (#2)Spool for data P=Production
  |_____ (#1)Form Name in EOS
```

- At our site, all referbacks are stored in a common file named EOSIALL. The DRPRINT routine will process this location searching for the referback named in the previous step (value #5). Once this member is located the record is parsed to locate the necessary AFP resources for ACIF. The routine handles records that span two lines. See the examples below:

```
//OPAPYCOM OUTPUT PAGEDEF=APYCOM,FORMDEF=APYCOM,FORMS=8889D
//OPCCWT60 OUTPUT PAGEDEF=CCWT60,FORMDEF=CCWT60,TRC=YES
//OPCCW023 OUTPUT PAGEDEF=CCW023,FORMDEF=CCW023,TRC=YES,
//         CHARS=(GS10,GS15,OB10,LR00)
```

- The routine can be invoked many ways where we will pass it two parameters. The first value is the form we need to print and the optional second parameter is the version number. If we pass only one value the version number defaults to "0" for the most recent document file. See the example below:

```
EXEC 'OD4.JCL.CNTL(DRPRINT)' 'AY2E 2'
                     |           |   |_____ Version –2
                     |           |_____ EOS form name
                     |_____ Routine to run
```

Build and Submit JCL

Review the sample JCL below as this is an example of how the JCL will be built and submitted. Values in **RED** would be created during the DRPRINT function.

```
//*===================================================
//RESTORE EXEC RDSPRINT,LOG='H',RINDX='RINDX'
//MINDX DD DISP=SHR,DSN=SYS2.EOS.PROD.AMI
//RSDARQ DD DSN=&&ARQ,DISP=(,PASS),UNIT=3390,
// SPACE=(CYL,(10,0))              >>> NO EXTENTS
//RSDPARQ DD DSN=&&PARQ,DISP=(,PASS),UNIT=3390,
// SPACE=(CYL,(10,10))             >>> TEMPORARY DS
//RSDARUPD DD DSN=&&ARUPD,DISP=(,PASS),UNIT=3390,
// SPACE=(CYL,(10,10))             >>> TEMPORARY DS
//S0000010.RSDPROF DD DISP=SHR,
// DSN=SYS2.EOS.PROD.PROFILE.MD001
//REPRDATA DD DSN=&&REPRINTD,DISP=(NEW,PASS),
// SPACE=(CYL,(100,200),RLSE),UNIT=SYSDA,
// DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VBM,DSORG=PS)
//S0000010.SYSIN DD *
 INIT ARQ
 ACCEPT RC=04
 PRINT REPORTINDEX,O=REPRDATA,RESTORE=Y,PRTREP=Y,
  USER=(BIN 253),VERS=-2,FORM=AY2E
//*
//*=====================================================*
//*            CREATE THE MOD:CA FILE                   *
//*=====================================================*
//ACIFSTEP  EXEC  PGM=APKACIF,REGION=3M
//STEPLIB   DD    DISP=SHR,DSN=SYS1.SAPKMOD1
//INPUT     DD    DISP=SHR,DSN=*.RESTORE.S0000010.REPRDATA
//OUTPUT    DD  DSN=&&OUTPUT,DISP=(NEW,PASS,DELETE),
//      SPACE=(32760,(150,150),RLSE),UNIT=SYSDA,
//      DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VBM,DSORG=PS)
//INDEX     DD  DSN=&&INDEX,DISP=(NEW,PASS,DELETE),
//      SPACE=(32760,(15,15),RLSE),UNIT=SYSDA,
//      DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VBM,DSORG=PS)
//RESLIB    DD  DSN=&&RESLIB,DISP=(NEW,PASS,DELETE),
//       SPACE=(12288,(15,15),RLSE),UNIT=SYSDA,
//      DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VBM,DSORG=PS)
//SYSPRINT  DD    SYSOUT=*
//SYSUDUMP  DD    DUMMY
//SYSIN     DD    *
CC = YES               /* USING CARRIAGE CONTROL     */
CCTYPE = M             /* TYPE OF CARRIAGE CONTROL    */
TRC = NO
INPUTDD = INPUT
OUTPUTDD = OUTPUT
PDEFLIB = SYS2.PSF.PROD.PDEFLIB
PAGEDEF = P1APYCOM
FDEFLIB = SYS2.PSF.PROD.FDEFLIB
FORMDEF = F1APYCOM
FONTLIB = SYS2.PSF.PROD.FONT300,SYS3.FONT300,SYS1.FONTLIBB
PSEGLIB = SYS2.PSF.PROD.PSEG300
OVLYLIB = SYS2.PSF.PROD.OVER300
USERLIB = OD4.AFPLIB
RESOBJDD = RESLIB
RESFILE = SEQ
```

```
 RESTYPE = ALL
 CPGID = 500
/*
//*        ADD RESOURCES TO AFP FILE FOR DR WORK
//PRINT EXEC PGM=IEBGENER
//MYOUT OUTPUT FORMS=8889D,DEST=CARY
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSUT1   DD DSN=*.ACIFSTEP.INDEX,DISP=SHR
//         DD DSN=*.ACIFSTEP.RESLIB,DISP=SHR
//         DD DSN=*.ACIFSTEP.OUTPUT,DISP=SHR
//SYSUT2   DD SYSOUT=9,OUTPUT=*.MYOUT
```

Description of the Job Steps listed above:

RESTORE – this is the EOS operation to automatically restore the needed data if it has been archived. This will also extract the data into a temporary dataset for later processing. The example shows the data we need is Version –2, although during a disaster we might need the most recent copy or Version 0.

ACIFSTEP – this step will process the data extracted by the restore step and create the MO:DCA format output file. The various AFP resource names have been filled-in by the routine.

PRINT – this step will concatenate the files created during the ACIFSTEP to cause the resources to be embedded at the front of the output file. The Sysut2 statement defines the output to deliver this data with resources to a remote destination of CARY using the appropriate paper stock.

System requirements and dependencies

- The DRPRINT routine was built using archival system EOS as the model. The design may work with other system archival tools if the functions are similar.
- If the archival tool does not have a collection area for referback information, this data could be added to the form table. This would be a manual process and requires updating when applications change.
- The person invoking this routine must have EOS authority to perform the restore operation of requested data - and the ARQ function.
- The person invoking this routine must have authority to submit jobs to the internal reader.
- There is minor customizing that needs to be done to identify site specific values. Change variables of:
  - Log_file – where should the log be written
  - DR_forms – location of the forms list. Follow example for columns.
  - EOS_Info – where is master referback listing
  - Destid – where should print file be sent