

---

## Appendix A. Output queue monitor

Some of the formatting tools described in this document may require you to change the attributes of your printer files. For example, if you want to use the form definitions and page definitions created using Infoprint Designer for iSeries or PPFA, you must use printer files that are defined with DEVTYPE(\*LINE). Other parameters in the printer file also must be changed, such as the FORMDF and PAGDFN parameters.

Unfortunately, many customers may not be in a position to make these changes for any of the following reasons:

- They do not have access to the source code from their vendor.
- They use the same printer file (often QSYSPRT) for all applications.
- They are concerned about having the changes overwritten if their vendor ships a new release of their software.
- They do not have the programming skills to make the changes.

One solution is to build an output queue monitor. This document provides guidance on how to do this.

---

### A.1 How an output queue monitor works

The spooled files continue to be generated by the application untouched. Each output queue you want to monitor has an entry in its definition for the DTAQ parameter. This points to an existing data queue object. Multiple output queues may point to the same data queue. Any time a spooled file is placed in the output queue in a RDY state, or if its state changes to RDY, an entry is added to the data queue with the spooled file name, job identifier, and output queue name.

The output queue monitor, a never-ending batch job, constantly checks for information arriving in the data queue. When it finds a new entry, the logic of the program can determine what to do with that spooled files and take appropriate action.

Do not use an output queue that is associated with an active printer writer for monitoring. You do not want conflicts between the writer trying to print your spooled file while the monitor attempts to perform other processing on it.

#### A.1.1 Basic logic of an output queue monitor

Your output monitor likely requires a combination of CL and some other high level language, such as RPG. Use CL commands to perform the object manipulation, such as changing spooled file attributes. The RPG, or other language programs, may be used for the table lookups or to call system API programs.

Assuming you have SCS spooled files, and you cannot modify the way they are created, you may want to use Infoprint Designer to enhance your output. This would be the basic logic to use in your monitor programs:

1. Use System API QRCVDTAQ to read the record from the data queue and to identify the spooled file you want to process. For more information about using

QRCVDTAQ, refer to the iSeries Information Center at:

<http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html>

To learn more about using the data queue reference in an output queue, and the layout of the information within it, see *iSeries Printer Device Programming*, SC41-5713.

2. Perform a table or file lookup to find out how that file should be handled. For example, what is the destination output queue and the name of the form definition and page definition objects that you created using Infoprint Designer that are to be used?

In some cases, enough information is provided from the data queue entry, such as the spooled file name, job or user name, or output queue, to uniquely determine the resulting action. In other cases, you must call system API QUSRSPLA to Retrieve Spooled File Attributes to obtain more information on which to make the decision. The information retrieved about the spooled file includes data stream type, page size, overlay name, user data, and form type, among other things. For more information about QUSRSPLA, see *AS/400 System API Reference*, SC41-5801, or the iSeries Information Center on the Web.

3. Use the Copy Spooled File (CPYSPLF) command to copy the spooled file to a physical file. To preserve the page and line spacing of the original spooled file, you must capture the carriage control information in the first byte of the file. To do this, make sure you specify `CTLCHAR(*FCFC)` in your CPYSPLF command, and make sure the target physical file record length is one more than the longest record in the spooled file.
4. Create or modify a printer file based on the requirements of the application. In the case of Infoprint Designer applications, you must specify `DEVTYPE(*LINE)`, form definition name (`FORMDF`), page definition name (`PAGDFN`), and target output queue (`OUTQ`). You must also specify `CTLCHAR(*FCFC)` so it knows to use the first byte as the carriage control information rather than as data.

If the form definition you use specifies duplex or different paper sources, you also must specify `DUPLEX(*FORMDF)` or `DRAWER(*FORMDF)`.

**Note:** The newly generated spooled file ends up being “owned” by the user who started the monitor program rather than the original owner. Therefore, the original owner is not able to perform a WRKSPLF with the default parameters to see their new files. One suggestion is to pick up the original user’s name from the job information that is picked up from the data queue and place it in the USRDTA field of the printer file.

5. Copy the data from the physical file to this new printer file using CPYF. This automatically creates a new spooled file.
6. Delete the original spooled file or move it to a hold or done queue.

Along with the monitor program, a robust system must also include housekeeping functions, such as error checking and table maintenance. If there is a problem, and the monitor ends abnormally, spooled files may have to be held and released to put a record back in the data queue.

---

## A.2 Other examples of use

Output queue monitors are not specific to the conversion of \*SCS to \*LINE. They can be used any time you require processing on a spooled file external to the original application that created it.

### A.2.1 Monitor for sending spooled files to AIX

You can use an output queue monitor to send spooled files to another machine where the destination, or other options, are variable- or parameter-driven. Appendix H of *AS/400 Printing Redbook V*, SG24-2160, describes using an output queue monitor for sending AS/400 spooled files to an Infoprint Manager for AIX server.

### A.2.2 Supplementing the APU monitor function

Advanced Print Utility (APU) uses a built-in output queue monitor to automate the reformatting of the spooled files. Each entry in the Monitor Action table contains selection criteria based on the output queue and the spooled file. However, if there are a large number of different unique spooled file designs to be sent to a large number of printers, the number of entries that you require in the APU monitor table may become unmanageable. A custom monitor, which selects the print definition and the destination queue independently, provides for a more manageable solution.

---

## A.3 Sample monitor shell program

This section contains a listing for a shell that may be used as a basis for your own monitor program. The mainline program shown here is CVTS0021. It calls CVTS0022 to perform the file manipulation. You must create CVTS0022 to suite your own requirements, such as conversion from SCS to LINE, including any table look-ups to determine the appropriate form definition and page definition names.

You must also add additional instructions in the ERROR HANDLING section.

### A.3.1 Monitor shell program listing

```

/*                                                                    */
/* *****                                                                    */
/* *                                                                    * */
/* *   PROGRAM   CVTS0021                                                                    * */
/* *                                                                    * */
/* *   By       Roger Drolet   rdrolet@ca.ibm.com                                                                    * */
/* *   DATE     May 1999                                                                    * */
/* *                                                                    * */
/* *   @ COPYRIGHT IBM CANADA 1999                                                                    * */
/* *   This software (source code and compiled objects)                                                                    * */
/* *   is the property of IBM Canada Ltd.                                                                    * */
/* *   No guarantees or warranties are made or implied.                                                                    * */
/* *   The user is expected to carry out testing to his                                                                    * */
/* *   or her own satisfaction.                                                                    * */
/* *****                                                                    */
/*                                                                    */
/*                                                                    */

```

```

/*      CVTS0021 - READ DTAQ AND CONVERT SPOOL FILE      */
/*      */
/*      This Program:      */
/*      - Wait up to 300 seconds to receive an entry from */
/*      the data queue      */
/*      - If no entry read, it check if the program should end */
/*      The program stop if:      */
/*      - Job was cancel with a ENJOB command      */
/*      - Subsystem was stop ENDSBS command      */
/*      - System shut down PWRDWN SYS command      */
/*      - If the entry is "STOP" program stop immediately */
/*      - Split the entry in fields      */
/*      - Call program CVTS0022 to do the spool file conversion */
/*      - Move or Delete the spool file      */
/*      - Return wait for an entry      */
/*      */
/*-----*/
/*      PROGRAM CHANGES      */
/*      */
/*      VERSION   DATE   PROGRAMMER  DETAIL      */
/*      YY/MM/DD      */
/*      1.0       99/05   RDROLET     INITIAL VERSION      */
/*      */
/*-----*/

          PGM

          DCL          &DTAQNNAME  *CHAR   10  VALUE(CVTSDTAQ)
          DCL          &DTAQLIB   *CHAR   10  VALUE(*LIBL)
          DCL          &ENTLEN    *DEC    5   VALUE(128)
          DCL          &ENTRY     *CHAR  128
          DCL          &WAIT      *DEC    5   VALUE(300)
          DCL          &ENDSTS    *CHAR    1

          DCL          &SPLFIL    *CHAR   10
          DCL          &JOBNAM    *CHAR   10
          DCL          &JOBUSR    *CHAR   10
          DCL          &JOBNBR    *CHAR    6
          DCL          &SPLNBRB   *CHAR    4   /* BINARY   */
          DCL          &SPLNBRD   *DEC    6   /* DECIMAL  */
          DCL          &SPLNBR    *CHAR    6   /* CHARACTER */
          DCL          &OUTQN     *CHAR   10
          DCL          &OUTQL     *CHAR   10
          DCL          &USRDTA    *CHAR   10

/*-----*/
/* RECEIVE AN ENTRY FROM THE DTAQ (FIRST IN FIRST OUT)      */
/*-----*/
RCVDTAQ:   CALL          PGM(QRCVDTAQ) PARM(&DTAQNNAME &DTAQLIB +
          &ENTLEN &ENTRY &WAIT)

/*-----*/
/* CHECK IF AN ENTRY WAS RECEIVE OR TIMEOUT      */
/*-----*/
          IF          COND(&ENTLEN *EQ 0) THEN(GOTO CMDLBL(TIMEOUT))

```

```

/* ----- */
/* CHECK IF THE ENTRY RECEIVE IS THE "STOP" COMMAND */
/* ----- */
          IF          COND(%SST(&ENTRY 1 4) *EQ 'STOP') THEN(GOTO +
                          CMDLBL(END))

/* ----- */
/* READ THE FIELD FROM THE RECEIVE ENTRY */
/* ----- */
          CHGVAR      &JOBNAM  VALUE(%SST(&ENTRY 13 10))
          CHGVAR      &JOBUSR  VALUE(%SST(&ENTRY 23 10))
          CHGVAR      &JOBNBR  VALUE(%SST(&ENTRY 33 6))
          CHGVAR      &SPLFIL  VALUE(%SST(&ENTRY 39 10))
          CHGVAR      &SPLNBRB VALUE(%SST(&ENTRY 49 4))
          CHGVAR      &SPLNBRD VALUE(%BIN(&SPLNBRB))
          CHGVAR      &SPLNBR  &SPLNBRD

/* ----- */
/* CALL PROGRAM CVTS0022 TO WORK WITH THE SPOOL */
/* ----- */
          CALL        PGM(CVTS0022) PARM(&ENTRY &OUTQN &OUTQL +
                          &USRDTA)

/* ----- */
/* Move or Delete the spool file */
/* ----- */
          IF          COND(&OUTQN *EQ '*DELETE') THEN(DLTSPFL +
                          FILE(&SPLFIL) +
                          JOB(&JOBNBR/&JOBUSR/&JOBNAM) +
                          SPLNBR(&SPLNBR) SELECT(*ALL))
          ELSE        CMD(CHGSPLFA FILE(&SPLFIL) +
                          JOB(&JOBNBR/&JOBUSR/&JOBNAM) +
                          SPLNBR(&SPLNBR) SELECT(*ALL) +
                          OUTQ(&OUTQL/&OUTQN) USRDTA(&USRDTA))

/* ----- */
/* GO READ NEXT ENTRY IN THE DTAQ */
/* ----- */
          GOTO        CMDLBL(RCVDTAQ)

/* ----- */
/* TIME OUT (CHECK IF THE JOB MUST END) */
/* ----- */
TIMEOUT:    RTVJOBA    ENDSTS(&ENDSTS)
          IF          COND(&ENDSTS *EQ '1') THEN(GOTO CMDLBL(END))
          ELSE        CMD(GOTO CMDLBL(RCVDTAQ))

          GOTO        CMDLBL(RCVDTAQ)

/* ----- */
/* ERROR HANDLING */
/* ----- */
ERROR:     GOTO        CMDLBL(RCVDTAQ)

END:       ENDPGM

```

