

# Automate journal receiver deletion

## Use an exit program to simplify journal management

Paul Reinschmidt

July 15, 2014

This article discusses how to automate the deletion of journal receivers to reduce direct access storage device (DASD) consumption and reduce backup time. An example exit program for the QIBM\_QJO\_DLT\_JRNRCV exit point to allow receiver deletion and days after being saved is presented along with instructions on using the registration facility to associate the exit program with the exit point.

### Let the system do it!

Are you tired of having to manually delete unneeded journal receivers from your system? Don't have time to write a program to search through your libraries and identify those journal receivers that are no longer needed for recovery and delete them? Don't have a high-availability Business Partner to manage journal receivers for you? Wouldn't it be nice to have the system automatically keep a certain number of days' worth of saved receivers around before deleting them? IBM i has the facilities to help you manage journal receivers automatically, thereby reducing consumption of DASD and shortening the time it takes to back up your system.

In this article, we'll take a look at an approach to automate the deletion of unneeded journal receivers by using the Delete Receivers option on the CHGJRN or CRTJRN commands in conjunction with the QIBM\_QJO\_DLT\_JRNRCV registered exit point. A sample ILE C program is provided to illustrate one way to make use of this exit point to govern how many days' worth of saved receivers should be kept around before they are deleted.

For more information to help you decide whether system-managed journal management is right for you, see the IBM® Redbooks® [Striving for Optimal Journal Performance on DB2 Universal Database for iSeries](#), especially pages 87 to 89.

### Preparing journal receivers for automatic deletion

In order to let the system automatically change and delete journal receivers, the *Manage receivers* and *Delete receivers* options on the Change Journal (CHGJRN) command should be set to

\*SYSTEM and \*YES, respectively, as shown in Figure 1. This allows the system to manage the attachment of new journal receivers and the deletion of unnecessary receivers. In addition to specifying these two options on the CHGJRN command, it is recommended that you specify a journal message queue to receive any messages associated with the journal. This will keep messages related to the deletion of journal receivers out of the QSYSOPR message queue and put them in a convenient place.

**Figure 1. Change journal (CHGJRN) options**

```
Journal message queue . . . . . JMSGQ1      Name
Library . . . . . JOURNALS      Name, *LIBL, *CURLIB
Manage receivers . . . . . *SYSTEM    *SYSTEM, *USER
Delete receivers . . . . . *YES      *NO, *YES
```

Another option to consider customizing on the CHGJRN command is 'Delete receiver delay time.' The default value is 10 minutes and this means that the system will keep trying to delete the journal receiver every 10 minutes and will put out a message every time it is unable to do so. This can produce a lot of unwanted messages in the journal's message queue. For example, in the example exit program shown in Listing 1, we have written the program to disallow the deletion of a journal receiver until 5 days have passed since it was saved. To avoid filling up the message queue with hundreds of messages per journal receiver over that time interval indicating that the receiver was ineligible for deletion, we have increased the delay time to 1440 minutes, or one daily attempt. You can set this delay time to whatever value you prefer in the range of 1 minute to 1440 minutes.

**Figure 2. Delete receiver delay time option (CHGJRN)**

```
Manage receiver delay time . . . 10      1-1440
Delete receiver delay time . . . 1440   1-1440
```

You can find these same options on the Create Journal (CRTJRN) command if you are creating a new journal for which you need the system to manage the deletion of journal receivers. Refer to the IBM Knowledge Center for information on these commands including possible restrictions on changing these options.

## Writing an exit program

The registered exit point (QIBM\_QJO\_DLT\_JRNRCV) and the exit point format (DRCV0100) are documented in IBM Knowledge Center. Refer to the topic, [Delete Journal Receiver Exit Program](#)

We have provided the program in Listing 1 as a sample to illustrate one approach to using an exit point program to determine whether a journal receiver is eligible for deletion. Before deleting the journal receiver, the system calls any user exit programs registered on this exit point for an indication as to whether the deletion should proceed. An exception to this is when the user has specified the \*IGNEXITPGM option on the DLTOPT parameter of the Delete Journal Receiver (DLTJRNRCV) command.

The sample exit program in Listing 1 creates the structures needed for the input and output parameters of the QIBM\_QJO\_DLT\_JRNRCV exit point, the QjoRetrieveJournalInformation

application programming interface (API), and the call to CEEOCT. Next, it copies the exit point input parameter into a structure and extracts the journal receiver name and library from the input and constructs the qualified journal receiver name needed in the call to the QJoRetrieveJournalInformation API. The API is used to obtain the date when the journal receiver was saved. The call to CEEOCT yields the local system date. We are interested in the Gregorian date returned by the API and we manipulate it slightly to get it in the same format as the journal receiver save date. After converting the dates to integers, the program subtracts the saved date from the current date and then applies the criteria to determine whether to allow the deletion to proceed. The two criteria chosen for this sample program are: the receiver must have been saved and the difference between the saved date and the current date must be greater than 5 (that is, the value of the constant numdays.) Otherwise, the program tells the system that the receiver is not eligible for deletion.

## Listing 1. Delete journal receiver exit program

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <recio.h>
#include <qjournal.h>
#include <leawi.h>

/* Begin main function */
main(int argc, char *argv[])
{
    char *temp_data;
    struct Qjo_RRCV0100 *rcv_info;
    long int rcv_info_length = 512;
    char qual_rcv_name[20];
    char format_name[8];
    typedef struct{
        long int exitpt_len;
        char exitpt_name[20];
        char exitpt_fmt[8];
        char jrnrcv_name[10];
        char jrnrcv_lib[10];
        char jrn_name[10];
        char jrn_lib[10];
        char Call_Sys_Job[1];
        char Call_IPL[1];
        char Proc_End[1];
        char jrn_type[1];
        char rmtjrn_type[1];
        char Save_Status[1];
        char Partial_Status[1];
        char detach_time[13];
    } Qjo_Dlt_JrnRcv_Info_DRCV0100_t;
    typedef _Packed struct Qjo_Dlt_JrnRcv_Status{
        long int Statusinfo_Length;
        char Dlt_Status[1];
    } Qjo_Dlt_JrnRcv_Status_t;
    Qjo_Dlt_JrnRcv_Info_DRCV0100_t input;
    Qjo_Dlt_JrnRcv_Status_t output;
    _INT4 lilian; /* Required parm not used in this program */
    _FLOAT8 seconds; /* Required parm not used in this program */
    _CHAR17 gregorian;
    char sv_dt_tm[13];
    char savdat[9];
    char curdat[9];
    int savdat_int;
    int curdat_int;
```

```

int result;
const int numdays = 5; /*Number of days of receivers to keep*/

/* copy input parm into structure */
memcpy(&input, (Qjo_Dlt_JrnRcv_Info_DRCV0100_t *)argv[1],
       sizeof(Qjo_Dlt_JrnRcv_Info_DRCV0100_t));

/* construct qualified journal receiver name from input */
memcpy(qual_rcv_name, input.jrnrcv_name,
       sizeof(input.jrnrcv_name));
memcpy(&qual_rcv_name[10], input.jrnrcv_lib,
       sizeof(input.jrnrcv_lib));

/* get space for output */
temp_data = (char *)calloc(1, sizeof(rcv_info_length));

/* Prepare the input parameters for the call to
   Qjo_JrnReceiverInformation */
rcv_info = (struct Qjo_RRCV0100 *)temp_data;

/* Call the API */
QjoRtvJrnReceiverInformation(
    rcv_info,
    &rcv_info_length,
    qual_rcv_name,
    "RRCV0100",
    NULL);

/* Get local date */
CEELOCT( &lilian, &seconds, gregorian, NULL );

/* Format the date data for the compare */
memcpy(&sv_dt_tm, rcv_info->Saved_Date_Time, sizeof(sv_dt_tm));
memcpy(savdat, &sv_dt_tm[1], sizeof(savdat));
savdat[8] = '\\0';
memcpy(curdat, &gregorian[2], sizeof(curdat));
curdat[8] = '\\0';
savdat_int = atoi(savdat);
curdat_int = atoi(curdat);
result = curdat_int - savdat_int;

/* Apply criteria for deletion of receiver. Receiver has to
   have been saved at least numdays ago */
if (input.Save_Status[0] == '1' && result > numdays){
    output.Statusinfo_Length =5;
    strncpy(output.Dlt_Status, "1",1);
    memcpy(argv[2], (Qjo_Dlt_JrnRcv_Status_t *)&output, 5);
}
else {
    output.Statusinfo_Length =5;
    strncpy(output.Dlt_Status, "0",1);
    memcpy(argv[2], (Qjo_Dlt_JrnRcv_Status_t *)&output, 5);
}
free((void *)temp_data);
} /* End of main function */

```

## Registering the exit program

After the exit program has been written and compiled, you can use the registration facility to associate it with the exit point. Use the Work with Registration Information (WRKREGINF) command to show a list of exit points defined on your system and page down until you find the QBM\_QJO\_DLT\_JRNRCV exit point as shown in Figure 3. Enter option 8 on that entry point.

### Figure 3. Work with registration information

```

_  QIBM_QJO_CHG_JRNRCV  CRCV0100  *YES  Change Journal Receiver
8  QIBM_QJO_DLT_JRNRCV  DRCV0100  *YES  Delete Journal Receiver

```

On the Work with Exit Programs screen, enter option 1 along with the name and library of your exit program (as shown in Figure 4).

### Figure 4. Work with exit programs: adding an exit program

```

                                Work with Exit Programs

Exit point:  QIBM_QJO_DLT_JRNRCV      Format:  DRCV0100

Type options, press Enter.
  1=Add   4=Remove   5=Display   10=Replace

                                Exit
                                Program
Opt      Program      Exit      Library
  1      DLTRCVP2      PAULR

```

## Summary

In this article, we have looked at an approach to automate the deletion of journal receivers using system management of journal receivers in conjunction with a QIBM\_QJO\_DLT\_JRNRCV exit point program. We saw how to change or create journals to allow the system to manage the deletion of receivers and looked at the particular options that need to be specified on the CHGJRN or CRTJRN commands. Next, an example of an exit program was presented and discussed to illustrate a way to establish criteria to govern the circumstances under which the system is allowed to delete journal receivers. In our example, we chose only to delete saved receivers that have a save date older than 5 days from the current date. Finally, we looked at how to associate the exit program with the QIBM\_QJO\_DLT\_JRNRCV exit point to tell the system which exit program to call before deleting a receiver.

## Resources

- [IBM i documentation for IBM Knowledge Center](#)
- [Striving for Optimal Journal Performance on DB2 Universal Database for iSeries](#)

© Copyright IBM Corporation 2014

([www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml))

Trademarks

([www.ibm.com/developerworks/ibm/trademarks/](http://www.ibm.com/developerworks/ibm/trademarks/))