

Journaling – Journal Receiver Diet Tip 1: Eliminating Open and Close Journal Entries, TIPS0607

Abstract

Not every journal entry that flows into a journal receiver is essential information. In fact, in some shops, certain journal entries are viewed as clutter. For example, journaling has the ability to optionally track every time your database physical file or stream file has been opened or closed and, by default, often does so. However, applications that perform many of these operations can fill the journal receivers with these journal entries, which are generally not needed.

This Technote gives a brief overview of the open and close journal entries. It also provides the steps for you to analyze your journal receiver to determine how much space is being consumed (if any) by these journal entries.

Written by Chris Kunding
System i High Availability -
SLIC Journal and Commitment Control

Contents

What is the benefit of having these optional journal entries?

The optional open and close journal entries can be used to perform analysis or debug problems. Debugging applications to determine who opened a file and to learn why an application is running slow is the main reason that people can benefit from capturing open and close entries. In comparison to simple database inserts, updates, and deletes of individual rows, the act of performing a full file close is a relatively expensive operation. An application that repeatedly performs this operation consumes a lot of resources. Looking into the journal receiver and seeing open and close journal entries can sometimes explain the overhead introduced by costly full close operations.

Having open and close journal entries helps with debugging or analyzing data, but are you debugging all the time? Hopefully not. Hence, most of the time these journal entries can be omitted. What if you decide later that you need them? You can turn the feature back on.

How much space can you save by omitting the presence of open and close journal entries?

To discover how much space you can save by omitting these optional entries, first display your journal to an outfile with the following CL command:

```
DSPJRN JRN(lib/jrn) OUTPUT(*OUTFILE) OUTFILE(lib/outfile)
```

Next run the following SQL statement to determine the number of bytes consumed by open and close entries:

```
SELECT SUM(JOENTL) FROM lib/outfile WHERE ((JOENTT = 'OP' OR JOENTT = 'CL') AND JOCODE = 'F') OR ((JOENTT = 'OF' OR JOENTT = 'CS' OR JOENTT = 'FC') AND JOCODE = 'B')
```

This select statement sums the entry lengths for the following journal entry code and type combinations.

Journal entry code	Journal entry type
F (Database File)	OP (open)
F “	CL (close)
B (Integrated File System)	OF (open)
B “	CS (close)
B “	FC (force)

You should receive a number of bytes that result from the previous SQL statement. However, if this SQL statement returns a dash or empty set, then you do not have any open or close journal entries and your journal receiver must already be following this journal diet plan.

Use the following SQL statement to gain an idea of the total number of bytes that are analyzed:

```
SELECT SUM(JOENTL) FROM lib/outfile
```

Calculate the total percentage of space that these entries consume by dividing the output from the first select statement result by the last (and multiply by 100).

How do you enable this feature?

The option to include or omit open and close entries is a parameter on the STRJRNPf or STRJRn CL command. However, omitting open and close journal entries is NOT the default. Hence, it is likely that many shops that never look for such performance problems collect such journal entries in vain. Prior to V5R3M0, it took both an ENDJRNPf (or ENDJRn) and a subsequent STRJRNPf (or STRJRn) OMTJRNE(*OPNCLO) command to eliminate such clutter. You can change this parameter without ending and restarting journaling as of release V5R3M0 by using the CHGJRNOBJ CL command.

Note: In contrast to the default behavior of the STRJRNPf and STRJRn CL commands, omitting open and close journal entries from your journal receivers is the default for SQL and when automatic journaling is used via the QDFTJRn data area.

Here is an example that demonstrates how to omit open and close journal entries:

```
CHGJRNOBJ OBJ((LIB/FILE *FILE)) ATR(*OMTJRNE) OMTJRNE(*OPNCLOSYN)
```

Conclusion

Open and close journal entries are not used for initial program load (IPL) recovery, APYJRnCHG, or RMVJRnCHG, nor are they required by High Availability business partners. However, these entries can sometimes be useful in better understanding your application's environment. You should now have a better understanding of the purpose behind these journal entries and know how much space you can save by eliminating them. Eliminating this space becomes especially attractive when you save journal receivers or send them to a remote journal.

Special Notices

This material has not been submitted to any formal IBM test and is published AS IS. It has not been the subject of rigorous review. IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends upon the customer's ability to evaluate and integrate them into the customer's operational environment.