JCC LogMiner Loader

Release Notes 3.6.2

May, 2020

From the JCC Toolset for Information Systems

Notices

All Rights Reserved

The information in this manual has been carefully checked and is believed to be accurate. However, changes to the product are made periodically. These changes are incorporated in new publication editions. JCC Consulting, Inc. may improve and/or change products described in this publication at any time. Due to continuing system improvements, JCC Consulting, Inc. is not responsible for inaccurate information that may appear in this manual. For the latest product updates, consult the JCC Consulting, Inc. web site at **www.jcc.com** or consult JCC in any of the ways indicated in the **"Contact Information" on page 2**. In no event will JCC Consulting, Inc. be liable for direct, indirect, special, exemplary, incidental, or consequential damages resulting from any defect or omission in this manual, even if advised of the possibility of such damages.

This product should not be used in any industry or fashion for which the underlying database products are not valid.

In the interest of continued product improvement, JCC Consulting, Inc. reserves the right to make improvements in this manual and the products it describes at any time, without notices or obligations.

This file and the LogMiner Loader software are confidential and proprietary to JCC Consulting, Inc. All documentation and the LogMiner Loader software are provided on an AS-IS basis.

Trademark Acknowledgments

JCC Toolset and JCC LogMiner Loader are trademarks of JCC Consulting, Inc. LogMiner, Oracle Rdb, Oracle 12 (and multiple other versions), MySQL, and Tuxedo are trademarks of Oracle Corporation. Windows (in its multiple versions) and SQL Server are trademarks of Microsoft Corporation. MariaDB is a trademark of MariaDB Corporation. Teradata is a

trademark of Teradata Corporation. DB2 is a trademark of IBM Corporation. OpenVMS is a trademark of Hewlett Packard Enterprise (HPE). OpenVMS is also available from VSI (VMS Software Inc.) which holds the copyright to versions created by VSI.

Uses in this Document

In this document, Oracle Rdb and Oracle's other RDBMS are referred to frequently and need to be distinguished. Consequently, Oracle Rdb is referred to as "Rdb" and Oracle's other database product (whatever its version) is referred to simply as "Oracle." Similarly, Oracle Rdb LogMiner is referred to as "Rdb LogMiner" or as "LogMiner." The JCC LogMiner Loader is sometimes referred to simply as "the Loader".

Disclaimer

This software is provided as is, without warranty of any kind. All express or implied conditions, representations and warranties, including any implied warranty of merchantability, fitness for particular purpose, or non-infringement, are hereby excluded to the extent permitted by applicable law. In no event, will JCC be liable for any lost revenue or profit or for special, indirect, consequential, incidental or punitive damages, however caused and regardless of the theory of liability, with respect to software made available here.

Contact Information

E-mail	JCC-LMLoader@JCC.com
Phone	+1 (740)587-0157
FAX	+1 (740)587-0163
Post Office	LogMiner Loader
	JCC Consulting, Inc.
	Box 381
	Granville, OH 43023

CHAPTER

Version 3.6.2

The notes in this chapter are specific to Version 3.6.2 of the JCC LogMiner Loader and represent changes made since Version 3.6.1.

The full documentation that was released with Version 3.6.0 has additional information on some of the topics discussed here.[1] The blogs - which are updated between releases - may also help you.[2]

Version 3.6.2 of the JCC LogMiner Loader includes the solution to a rare issue that can cause *loss of data*. It also includes solutions to two other issues that are rare, but awkward when they happen. Further, this version includes a way of providing Oracle credentials to the process to tell if its safe to run AIJ backup.

This chapter reports the details in the sections Corrected Issues and Enhancements.

1. Find the documentation for download in the upper right corner at http://www.jcc.com/lml.
2. Find the blogs at http://www.jcc.com/lml-blog.

Corrected Issues

All known issues are addressed. Some of the issues relate to only one target or to one specific configuration and are not relevant in all cases.

The issues addressed are:

1. Serious issue that could result in skipping a set of transactions if there is an exception that causes the Loader to fail and the Loader also fails on restart.
2. An access violation in processing of a particular complex FilterMap statement.
3. A global section error that was reported as a shared memory thread mismatch or an inability to use 32 Loader threads.
4. Failure to appropriately collect credentials (required for OCI targets) when running the procedure to check whether backup can be safely run.

SERIOUS ISSUE - Possible Data Loss

An obscure set of issues could cause a loss of data updates.

The situation occured if an exception that caused the Loader to fail, then caused the Loader to fail in restart on the first checkpoint with the same or a similar exception. The result was the loss of a checkpoint's worth of data changes.

This is an unusual set of circumstances, but the result is extremely serious.

This bug is fixed in Version 3.6.2.

Complex FilterMap Statement Caused Access Violation

If a column was referenced multiple times in a FilterMap statement, it could cause an access violation while processing the Control File. The exception message would resemble

```
Unhandled exception in dba_initialize: no such file or directory
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual address=000000000000
%TRACE-F-TRACEBACL, symbolic stack dump follows
```

For example, a FilterMap statement like this one would trigger the exception.

```
`FilterMap~SMPL_ROW~JCCLML_USERNAME<>'PURGE' \`
`    AND MY_NUM IS NOT NULL \`
`    AND PROD_DT IS NOT NULL  \`
`    AND SMPL_TYP IS NOT NULL \`
`    AND ( MY_NUM STARTING WITH 'R' OR MY_NUM STARTING WITH 'F' )`
```

The workaround was to reconstruct the FilterMap statement to only reference a variable once.

This bug is fixed in Version 3.6.2.

Loader Statistics

A bug introduced to support the STATE display for Loader statistics (JCC_LML_STATSTICS) caused exceptions with some runs of the statistics utility. These happened only when *not* using parallel threads.

The error was a result of improper mapping of the global section. It could appear in two ways.

1. An exception that appears as ‘shared memory thread mismatch’
2. An exception that appears as an access violation when monitoring a session with 32 Loader threads.

When using the Statistics utility, the exception message for the first of those looked like this:

```
%jcc_lml_statistics: Shared memory thread mismatch.
```

The workaround for this problem was to use threads, but declare a single parallel thread.

```
Parallel~1~1
```


The workaround for the problem with the 32nd thread was to limit the number of threads to 31 or fewer.

For either example, this bug is fixed in Version 3.6.2.

Checking the Safety of Running Backup

The intent of the procedure JCC_CLML_AIJ_BACKUP_SAFE is to determine whether the LogMiner has completed the backed up AIJ files and is processing in the live AIJ files, *before running AIJ Backup.* Failure to let the LogMiner process the backup files before running AIJ Backup can, in rare circumstances, cause the loss of multiple transactions.[3]

A careful approach is to run the procedure in the DCL to do backup and cause the backup to wait, if the procedure returns “False”.

A difficulty, with the careful approach, arose when the target was OCI (Oracle). The procedure to check whether it is safe to run backup uses another procedure called JCC_LML_DUMP_CHECKPOINT to determine how far the LogMiner has gotten. For OCI targets, checkpoint data is written to the database and the procedures can’t access the database without the proper credentials (username and password). The JCC_CLML_AIJ_BACKUP_SAFE procedure was not properly collecting and including these credentials when the procedure was included in a batch procedure.

A workaround was to copy the JCC_CLML_AIJ_BACKUP_SAFE procedure to a target specific environment and add the username and password to the batch procedure, after the call to JCC_LML_DUMP_CHECKPOINT.

For version 3.6.2, the issue is resolved by including the credentials in the parameter that names where to find the checkpoint information.

3. See http://www.jcc.com/lml-abs-bad in the JCC blogs for more on this Oracle bug. See Safety Test for AIJ Backup in the full documentation for the procedure to test for safety.

Syntax for JCC_CLML_AIJ_BACKUP_SAFE

The syntax for JCC_CLML_AIJ_BACKUP_SAFE is unchanged, except in how the parameter checkpoint name will be interpreted.

```
$ jcc_clml_aij_backup_safe          -
        <database name>             -
        <Loadername>                -
        <checkpoint name>           -
        [<checkpoint type>]
```

where

<database name>. Database name is the name of the source database. This parameter is not passed to JCC_LML_DUMP_CHECKPOINT when it is called.

<LoaderName>. Loadername provides the LoaderName for the instance which is being run.

<checkpoint name>. Checkpoint Name indicates where the checkpoint information is stored and includes the credentials as appropriate. It varies with checkpoint type. See “Name for the Checkpoint” on page 8 for the details.

<[checkpoint type]>. Type is optional and is the type of checkpoint stream. The options are LML_INTERNAL, OCI, and RDB. The default is LML_INTERNAL.

Syntax for JCC_LML_DUMP_CHECKPOINT

The syntax for JCC_LML_DUMP_CHECKPOINT is unchanged, except in how the parameter checkpoint name will be interpreted.

```
$ jcc_lml_dump_checkpoint           -
        <LoaderName>                -
        <checkpoint name>                       -
        [<checkpoint type>]
```

where

<LoaderName>. The first parameter provides the LoaderName for which the checkpoint is being run.

<Checkpoint Name>. Checkpoint Name indicates where the checkpoint information is stored and includes the credentials as appropriate. It varies with checkpoint type.

[<checkpoint type>]. Type is optional and is the type of checkpoint stream. The options are LML_INTERNAL, OCI, and RDB. The default is LML_INTERNAL.

Name for the Checkpoint

The checkpoint name parameter includes sufficient information to retrieve the checkpoint. Its format is dependent on the checkpoint type.

- When the checkpoint type is LML_INTERNAL, the parameter is the name of the checkpoint file.
- When the checkpoint type is RDB, the parameter is the name of the Rdb database that stores the checkpoint information.
- When the checkpoint type is OCI, the parameter provides the name of the Oracle database and the credentials to access that database. The format needed for that is
<username>@<database name>/<password>

Example of Parameters Used

If the procedure JCC_CLML_AIJ_BACKUP_SAFE is called with the parameters shown in green in the chart, it, in turn, calls the procedure JCC_LML_DUMP_CHECKPOINT with the parameters shown in blue on the last line of the chart.

TABLE 1. JCC_CLML_AIJ_BACKUP_SAFE calls JCC_LML_DUMP_CHECKPOINT

Database Name	Loadername	Checkpoint Name	Ckpt Type
reg_test_db	0112T0012OCI	011_2too12_1@regtest.jcc.com/o11_2too12_1	OCI
	0112T0012OCI	011_2too12_1@regtest.jcc.com/o11_2too12_1	OCI

The values shown for the parameters are the values used in one of the regression tests and are included in the examples below.

Note that the checkpoint type is OCI and the checkpoint name is composed of the the database name and the credentials in the format required

```
<username>@<database name>/<password>
```

Example of the New Support

The following example from the regression testing illustrates the new functionality, when the JCC_LML_DUMP_CHECKPOINT procedure is called directly.

```
$!
$!
$!Demonstrates that the new support for the target including
$!the username and password in the checkpoint name parameter.
$!
$ jcc_lml_dump_checkpoint O112TOO12OCI                      -
$        o11_2too12_1@regtest.jcc.com/o11_2too12_1 OCI

JCC LML Dump Checkpoint D03.06.02 (built 23-MAR-2020
17:22:50.34)

Connected to ORACLE(regtest.jcc.com) as user o11_2too12_1.

 -- Checkpoint restart information --

--- Parallel mode ---
Write Timestamp:     23-MAR-2020 15:41:13.50
LoaderName:          O112TOO12OCI
Completion Flag:     S
Checkpoint Interval: 15
Input Data Source:   LML_CONT_O112TOO12OCI
Last Transaction:
        Start Time:  23-MAR-2020 14:40:52.43
        Commit Time: 23-MAR-2020 14:40:52.44
        TSN:         3997468030
        LSN:         444326
        AERCP:       1-28-29156-25172-3997468030-3997468030
        RM TID:
```


Example of the Backwards Compatibility

The following example from the regression testing illustrates that the prior functionality is unchanged, when the JCC_LML_DUMP_CHECKPOINT procedure is called directly.

```
$!
$!
$!Demonstrates that the historical functionality is
$!preserved unchanged.
$!
$ jcc_lml_dump_checkpoint O112TOO12OCI regtest.jcc.com OCI

JCC LML Dump Checkpoint D03.06.02 (built 23-MAR-2020
17:22:50.34)

username: o11_2too12_1
password: o11_2too12_1
Connected to ORACLE(regtest.jcc.com) as user o11_2too12_1.

 -- Checkpoint restart information --

--- Parallel mode ---
Write Timestamp:     23-MAR-2020 15:41:13.40
LoaderName:          O112TOO12OCI
Completion Flag:     S
Checkpoint Interval: 15
Input Data Source:   LML_CONT_O112TOO12OCI
Last Transaction:
        Start Time:  23-MAR-2020 14:40:52.43
        Commit Time: 23-MAR-2020 14:40:52.44
        TSN:         3997468030
        LSN:         444326
        AERCP:       1-28-29156-25172-3997468030-3997468030
        RM TID:
```

Exception Message

The input below generates an exception because the checkpoint type is OCI and the name passed includes the username and database name in proper format, but lacks the slash and password.

```
$!
$ jcc_lml_dump_checkpoint O112TOO12OCI o11_2too12_1@regtest.jcc.com OCI
%jcc_lml_dump_checkpoint: Value for <name> is invalid Oracle format.
  Valid format is username@database/password
  Found 'o11_2too12_1@regtest.jcc.com'
```


CHAPTER

Version 3.6.1

The notes in this chapter are specific to Version 3.6.1 of the JCC LogMiner Loader and represent changes made since Version 3.6.

The full documentation that was released with Version 3.6.0 has additional information on some of the topics discussed here.[1] The blogs - which are updated between releases - may also help you.[2]

Version 3.6.1 of the JCC LogMiner Loader expands support for the Kafka output option, corrects all known issues, and provides an additional enhancement to analysis of performance.

This chapter contains the sections:

- Corrected Issues
- Enhancements

1. Find the documentation for download in the upper right corner at http://www.jcc.com/lml.
2. Find the blogs at http://www.jcc.com/lml-blog.

Corrected Issues

All known issues are addressed. Some of the issues relate to only one target or to one specific use and would not have been relevant in all cases.

The issues addressed are:

1. Statistics STATE display showing asterisks, in limited circumstances.
2. Heartbeat logging disabled in certain scenarios.
3. FilterMap definition could result in CLML ACCVIO stack dump.

State Display for Statistics Monitor

State was introduced with Version 3.6 of the JCC LogMiner Loader as one of the report types available with the Statistics Monitor. The State report provides detailed information on the state (status) of each thread. See the full 3.6 documentation for more information.

On rare occasions, the Loader statistics STATE display would show 7 asterisks in the place where the latency[3] should be displayed, such as

```
[*******] Output->Send message
```

This occurred in two situations. The first was when there was a timing issue and the calculated data would not fit into the allotted space. The second was when the data was rounded to the requested number of decimal places for display and the resulting value exceeded the allotted space.

These problems have been corrected.

3. Latencies are discussed later in this document.

Heartbeat Logging

If a database becomes quiescent with no update transactions for an extended period of time, the AIJ backup process will stall behind the Rdb LogMiner. The JCC LogMiner Loader provides a tool called “Heartbeat” to prevent this issue.

It may be desirable to see heartbeat activity in the log file. Heartbeat logging can be enabled by adding to the Control File the line

```
logging~heartbeat
```

Originally, enabling did not work as expected if the heartbeat table was also defined in the Control File or replicated to the target. The heartbeat feature continued to work as anticipated, but the results were not logged.

This problem has been corrected.

FilterMap Definition and Abnormal Results

In version 3.6.0 of the JCC LogMiner Loader, use of the FilterMap feature could result in the CTL process (JCC_RUN_CLM_LML) exiting with an access violation such as that shown.

```
FilterMap~TABLE_PLACE_MAP1~where COL1 between 500 and 999
%SYSTEM-F-ACCVIO, access violation, reason mask=04, virtual
address=000000000055C004, PC=FFFFFFFF84073CA0, PS=0000001B
%TRACE-F-TRACEBACK, symbolic stack dump follows
image module routine line rel PC abs PC
LIBRTL LIB$MALLOC LIB$VM_CALLOC 26235 0000000000006330 FFFFFFFF84073-
CA0
o
o
o
```

This reported problem was never reproduced in JCC testing. The resulting exception may vary by system or environment. This issue has, however, been fixed sufficient that, with version 3.6.1, it does not occur in the environment that previously produced it.

Enhancements

Two enhancements are added for Version 3.6.1 of the JCC LogMiner Loader. These are:

1. A Kafka Target Expansion.
2. A latency column for the Statistics CALLSTACK output.

Kafka Target Expansion

Support for Kafka as a target of the JCC LogMiner Loader was introduced with Version 3.6. At that time, the Kafka models supported were limited to "Transaction" and "ExactlyOnce". These are defined in the documentation for the Kafka option that was released at the same time as Version 3.6 of the Loader.

Release 3.6.1 adds Flush and Memory to the models available for use with the Kafka interface.

Flush

The Flush model ensures that all rows in a Loader checkpoint are flushed and verified as received on Kafka topic partitions before the Loader thread updates it's highwater checkpoint data.

Memory

The Memory model forces a Flush only if the Java Virtual Machine which the Loader uses to send data to Kafka is low on memory. In this case, the Loader updates highwater checkpoint data without verifying that the data has been received on Kafka topic partitions.

With Version 3.6.1, Transaction remains the default, but any of the models may be enabled with one of these lines in the Control File

```
Kafka~Model~Flush
```

```
Kafka~Model~Memory
Kafka~Model~ExactlyOnce
Kafka~Model~Transaction
```

Some users of the JCC LogMiner Loader have opted to use the Flush model. The choice is apparently made for Kafka performance reasons.

The Memory model should only be used in situations for which lost data is an acceptable option. For example, the Memory model might be a valid choice if the data is used to present realtime statistics information for visual consumption.

Latency Column Added to Statistics

Latency is the period of time from the start of an action to current moment or to the point at which that action is complete. Reporting latency and attributing latency to specific stages can be an important asset in analyzing performance.

See Loader Latency Reporting in the section on the Statistics Monitor in the chapter Monitoring an Ongoing Loader Operation in the full documentation.

The JCC LogMiner Loader has been equipped with more and more latency reporting as additional opportunities have been suggested. For Version 3.6.1, latency reporting is added to the call stack reporting available with the command
`JCC_LML_STATISTICS CALLSTACK`

The call stack reporting now includes a column that shows the latency. In the following example, the latency numbers are highlighted for clarity.

```
$ jcc_lml_statistics SELNERDBJDB callstack

JCC LogMiner Loader Statistics D03.06.01 (built 19-FEB-2019
13:07:01.84)
```



```
******* Thread 0 call stack ***************************
** [  49.1m] LML
** [   0.95]  Output
** [   0.95]   Output->Process buffer
** [7.001ms]    Output->Record
** [7.001ms]     Output->Message
** [7.001ms]      Output->Send message
** [7.001ms]       Write->JDBC
** [6.001ms]         PEOPLE_JDBC_TARGET[batch:20]
*****************************************************

Threads 1-31 inactive
```

See the “Time Scale Conversion” in the chapter Monitoring an Ongoing Loader Operation in the 3.6 documentation for an explanation of the time units shown.