SYBASE®
An SAP® Company

New Features Bulletin

Replication Agent™ 15.6

Linux, Microsoft Windows, and UNIX

DOCUMENT ID: DC00503-01-1560-01
LAST REVISED: November 2010

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

Replication Agent 15.6 New Features

Describes the new features that are available for Replication Agent™ 15.6 for Linux, Microsoft Windows, and UNIX.

Sub-capacity Licensing

You can license Replication Agent 15.6 for fewer than the total number of processors on a multiprocessor machine.

Subject to the terms of your software license agreement and product documentation, you may be able to license a subset of the processors available on a physical machine using the SySAM sub-capacity license functionality. See your software license agreement and product documentation to determine whether SySAM sub-capacity licensing is allowed and if so, what type of partition or resource allocation technologies are supported.

Oracle Support

Learn about the new Replication Agent 15.6 features for Oracle support.

Reload User Information

In versions earlier than 15.6, the only way to synchronize users in the primary database and users in the Replication Agent System Database (RASD) was to use **pdb_xlog init, force**. However, this command also refreshes the table schema and therefore requires extra time.

In Replication Agent 15.6, you can use the **ra_updateusers** command to reload all the user information from the primary database to the RASD. The **ra_updateusers** command refreshes only user information.

Note: This command is available only for Oracle.

Syntax

```
ra_updateusers
```

Usage

- **ra_updateusers** reloads user information to the RASD from the primary database. Use this command when user information in the RASD becomes unsynchronized with the primary database.
- When you invoke **ra_updateusers**, Replication Agent:
  1. Deletes all user information from the RASD

2. Queries the primary database for user information
3. Repopulates the RASD with the user information returned from the primary database

- Use **ra_updateusers** only when Replication Agent is in the Admin or Replication Down state.

Skip Log Reader Errors

You may need to prevent log reader errors from stopping replication in systems that cannot afford down time during periods of high volume and high demand. You can then address skipped errors at a more convenient time.

Use the **skip_lr_errors** parameter to skip log record processing errors and log warning messages to the Replication Agent system log without stopping replication.

Note: This parameter is available only for Oracle.

Default
false

Value
true – enables Replication Agent to skip log record processing errors and continue replication.

false – disables Replication Agent from skipping log record processing errors.

Comments

- If you configure **skip_lr_errors** to true, Replication Agent logs the log record processing error encountered and also logs a warning that the error has been skipped. If the transaction ID, operation ID and locator of the log record are available at the time of the error, these are also logged. Replication Agent continues processing transaction log records.
- If you configure **skip_lr_errors** to false, Replication Agent throws an exception, stops all replication processing, and transitions to the Replication Down state.
- **skip_lr_errors** is intended only for troubleshooting with assistance from Sybase® Technical Support.
- You can change **skip_lr_errors** only when Replication Agent is in the Admin or Replication Down state.

Warning! Use of this parameter does not guarantee that there will be no transaction loss, nor does it guarantee that the RASD is synchronized with the primary database when log record processing errors are skipped.

Oracle Database 11g Release 2

All of the functionality that Replication Agent supports for Oracle Database 11g Release 1 is also supported by Replication Agent for Oracle Database 11g Release 2. Replication Agent also supports some functionality introduced by Oracle Database 11g Release 2.

Replication Agent supports functionality common to both Oracle Database 11g Release 1 and Oracle Database 11g Release 2, including:

- Oracle DDL and DML replication in systems with and without Automatic Storage Management (ASM) and Real Application Clusters (RAC).
- Use of the Oracle Recovery Manager (RMAN) utility to truncate old archive log files.
- Use of the Oracle Recycle Bin and replication of Oracle Flashback operations.
- Oracle Data Guard.

Replication Agent also supports some features that are new as of Oracle Database 11g Release 2:

- Use of the **FORCE** option with a **CREATE OR REPLACE TYPE** statement on types with type dependencies.
- DDL statements on tables enabled for Flashback Data Archive.
- Version 11.2 time zone files and new time zone behavior.

Network Configuration File Location and Structure

The `tnsnames.ora` file is located in `ORACLE_HOME\network\admin`. In an Oracle Database 11g Release 2 instance running ASM or RAC, the `tnsnames.ora` file is read by default from the grid infrastructure home directory at `Grid_home\network\admin`.

If you are using an Oracle Database 11g Release 2 instance running ASM or RAC, set the Replication Agent **asm_tns_filename** parameter to `Grid_home\network\admin\tnsnames.ora`.

By default, the `tnsnames.ora` file at `Grid_home\network\admin` contains an incomplete ASM entry that lacks information in the DESCRIPTION and SERVICE_NAME fields. If you are using an Oracle Database 11g Release 2 instance running ASM, set the Replication Agent **asm_tns_connection** parameter to the ASM connection name specified in this incomplete ASM entry. Replication Agent completes the DESCRIPTION and SERVICE_NAME fields, and you can then use the `tnsnames.ora` file in `Grid_home\network\admin` to connect to the ASM instance server.

Time Zone File

By default, Oracle Database 11g Release 2 uses the large time zone file, `timezone_11.dat`. This file contains all the time zones defined in the database.

If you are using Oracle Database 11g Release 2, set the Replication Agent **pdb_timezone_file** parameter to the location of the `timezone_11.dat` file:

```
ra_config pdb_timezone_file, $ORACLE_HOME/oracore/zoneinfo/
timezone_11.dat
```

User-Defined Type Dependencies

You can use the **CREATE OR REPLACE TYPE** command to change the definition for an existing user-defined type. However, this command throws an error if the referenced type has table or type dependencies.

Oracle Database 11g Release 2 allows you to use **FORCE** with the **CREATE OR REPLACE TYPE** command to replace a type that has a type dependency:

```
CREATE TYPE mytype1 AS OBJECT (a number) NOT FINAL;

CREATE TYPE mytype2 UNDER mytype1 (b varchar(10));

CREATE OR REPLACE TYPE mytype1 FORCE AS OBJECT (c varchar(20));
```

Oracle Database 11g Release 2 does not allow you to use **FORCE** with the **CREATE OR REPLACE TYPE** command to replace a type that has a table dependency:

```
CREATE TABLE mytable1 (colA mytype1);

CREATE OR REPLACE TYPE mytype1 FORCE AS OBJECT (d number);
```

The last command results in an error because `mytype1` has a table dependency on `mytable1`:

```
ERROR at line 1:

ORA-22866: cannot replace a type with table dependents
```

Replication Agent supports use of the **FORCE** option with the **CREATE OR REPLACE TYPE** command in Oracle Database 11g Release 2 to replace types with type dependencies but not for types with table dependencies.

Flashback Data Archive Support for DDL Commands

Oracle Flashback allows database administrators and users to view past states of database objects and to restore database objects to a previous state without using point-in-time recovery. Users can query past data, query metadata to build a detailed history of changes, recover data to a previous point in time, and roll back transactions while the database is online.

Replication Agent supports the replication of DDL commands on tables that are being tracked with the Flashback Data Archive in Oracle Database 11g Release 2. These DDL commands include:

- **Add**, **Drop**, **Rename**, **Modify Column**
- **Drop**, **Truncate Partition**
- **Rename**, **Truncate Table**
- **Add**, **Drop**, **Rename**, **Modify Constraint**

Linux Block Devices

Block devices are disks or memory regions in which data is accessed in blocks. Linux block devices are generally accessed through I/O buffers. However, Linux block devices can also be accessed directly, and Oracle uses direct I/O to write to its redo logs. It is therefore important for Replication Agent to directly read from the block devices on which Oracle redo logs are stored. Otherwise, stale or nonexistent block data from device I/O buffers may interfere with replication.

Use the **lr_direct_read** parameter to indicate how Replication Agent reads Oracle redo logs.

Note: This parameter is available only for Oracle and is intended for Oracle data servers running on the Linux operating system.

Default
false

Values
true or false

Comments

- If **lr_direct_read** is true, Replication Agent reads Oracle redo log files directly on Linux, bypassing block device I/O buffers.
- Set **lr_direct_read** to true when the Oracle redo logs are stored on a Linux block device —as with the ASM Library Driver (ASMLib)—or an Oracle Cluster File System (OCFS) file.

Oracle Cluster File System 2 (OCFS2)

OCFS2 is a shared-disk cluster file system. OCFS2 is not a general-purpose file system but can contain all Oracle data files, redo and archive log files, and control files. OCFS2 presents a consistent file system image across the servers in an Oracle cluster and eliminates the need for managing raw devices, thereby simplifying the administration of RAC.

Sybase has tested and certified that Replication Agent can successfully replicate data from an Oracle RAC database running on OCFS2.

High Availability

A primary data server may employ a high-availability solution, such as failover clustering, to minimize downtime in the event of hardware or software failure.

Although Replication Agent does not provide any high-availability solutions, it works with a third-party high-availability solution for the primary database if:

- Replication Agent is installed on a shared file system, such as OCFS, a Network File System (NFS), or a Veritas Cluster Server (VCS). The Replication Agent binaries, configuration files, and RASD files must be installed on the system.
- A third-party cluster-management solution—such as Sun Cluster Manager, Veritas Cluster Manager, or Oracle Cluster Ready Services (CRS) —is used to automatically start Replication Agent in the event of failover.

ASMLib

Replication Agent for Oracle adds support for the ASMLib library for Linux.

ASMLib is a support library for Oracle ASM instances running on the Linux operating system. ASMLib reduces the start-up time for ASM instances and improves the performance of ASM disk groups. ASMLib also allows ASM disk management at the operating system level by allowing users to refer to ASM disks by logical names instead of by their physical names. For example, a query to the V$ASM_DISK view shows that an Oracle database without ASMLib contains these ASM disks:

```
SQL> select path from v$asm_disk;

PATH

-------------------------

/dev/oracleasm/disks/sdd

/dev/oracleasm/disks/sdc

/dev/oracleasm/disks/sdb

/dev/oracleasm/disks/sda
```

The disks are identified by physical names and paths. With ASMLib, these same ASM disks are instead identified by logical names and paths:

```
SQL> select path from v$asm_disk;

PATH

----------

ORCL:DVOL2

ORCL:DVOL1

ORCL:RVOL2

ORCL:RVOL1
```

The physical ASM names and disk paths are used for replication, and Replication Agent 15.6 supports Oracle ASMLib by resolving the physical names from the ASMLib logical names.

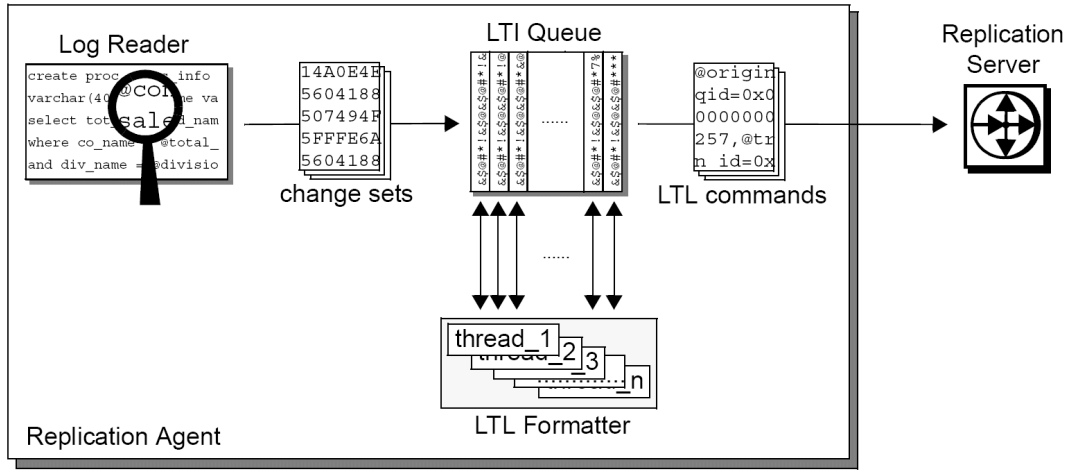
Note: ASMLib uses Linux block devices. For information on configuring Replication Agent to work with Linux block devices, see *Linux Block Devices* on page 5.

Parallel LTL Formatting

Replication Agent can now format log transfer language (LTL) commands in parallel, improving performance on machines with multiple processors.

The LTL formatter is an internal component of Replication Agent that converts change sets into LTL commands. In versions earlier than 15.6, the LTL formatter has been a performance bottleneck because items in the log transfer interface (LTI) queue could be processed only serially. Replication Agent 15.6 provides a multithreaded LTL formatter that processes items in the LTI queue in parallel. This may improve performance on machines with multiple processors.

Figure 1: Replication Agent LTL Formatting



Replication Agent 15.6 uses two configuration parameters for configuring the size of the LTI queue and the number of LTL formatter threads: **lti_max_buffer_size** and **lti_formatter_count**, which is a new parameter. Replication Agent 15.6 also provides new LTI statistics to assist you in performance and tuning.

Multiple LTL Formatter Threads

By adjusting the number of LTL formatter threads working in parallel on a machine with multiple processors, you can improve Replication Agent performance.

The **lti_formatter_count** parameter specifies the number of threads in the LTL formatter that work concurrently on items in the LTI queue. You should adjust the value of this parameter according to the number of processors available to Replication Agent in the machine on which it is installed.

Default
3

Values
1–200.

Comments

- Each thread specified by **lti_formatter_count** works on a separate item in the LTI queue.
- To determine if performance may be improved by increasing the value of **lti_formatter_count**, examine the LTI statistics for "Current number of commands in the LTI queue" and "Current number of unformatted commands in the LTI queue." When the number of commands in the LTI queue is at or near capacity while the number of unformatted commands is closer to capacity than to zero, increasing the value of **lti_formatter_count** may improve Replication Agent performance.
- You can change **lti_formatter_count** only when Replication Agent is in the Admin or Replication Down state.

LTI Queue Size

You can improve Replication Agent performance on a machine with multiple processors by adjusting the size of the LTI queue.

The **lti_max_buffer_size** parameter has been updated to specify the maximum number of items that can be stored in the LTI queue. Set this parameter according to the number of LTL formatter process threads and the number of processors available.

Default
5000

Values
1000–100000.

Comments

- The **lti_max_buffer_size** specifies the maximum number of items that the log reader can place in the LTI queue.
- Setting **lti_max_buffer_size** to a value that is too large may degrade performance if there is insufficient memory available.
- You can change **lti_max_buffer_size** only when Replication Agent is in the Admin or Replication Down state.

LTI Statistics

Replication Agent statistics help you observe the effect of adjusting configuration parameters for performance and tuning. Replication Agent 15.6 provides new statistics to observe the behavior of the LTI queue and the LTL formatter.

Table 1. New Log Transfer Interface Statistics

Component	Statistic	Description
LTI	LTI queue size	Current number of commands in the LTI queue
LTI	LTI unformatted command count	Current number of unformatted commands in the LTI queue

These statistics have been removed.

Table 2. Removed Log Transfer Interface Statistics

Component	Statistic	Description
LTI	Average data arrival time	Average time (in milliseconds) LTI waits between receiving change sets from Log Reader since last reset
LTI	Input queue size	Current number of change sets in the LTI input queue
LTI	Output queue size	Current number of distributes in the LTI output queue