



New Features Guide

Replication Server[®] Options

15.7.1 SP120

Linux, Microsoft Windows, and UNIX

DOCUMENT ID: DC01004-01-1571120-01

LAST REVISED: December 2013

Copyright © 2013 by SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices.

Contents

New Features in Replication Server Options 15.7.1	
SP120	1
New Features in Replication Agent for Oracle	1
Oracle XStream	1
Enhancements in Replication Agent for Microsoft SQL Server	18
Replicate Microsoft SQL Server Large Object Datatypes into an SAP HANA Database	18
Enhanced server_xlog Command	19
Replication Agent Security Enhancements	20
Support for Maximum-Strength Password Encryption	20
SAP JVM Support	21

Contents

New Features in Replication Server Options 15.7.1 SP120

In Replication Server® Options 15.7.1 SP120, the Replication Agent™ for Oracle now can also use XStream APIs to read from logs, Replication Agent for Microsoft SQL Server is enhanced to support large datatypes in heterogeneous environments, and the Replication Agent installer includes security enhancements, and support for the SAP® Java Virtual Machine (JVM).

New Features in Replication Agent for Oracle

In addition to Oracle LogMiner, you can now alternatively configure the Replication Agent for Oracle 15.7.1 SP120 Log Reader component to use Oracle XStream APIs to read from the transaction logs.

Oracle XStream

The Replication Agent for Oracle Log Reader component uses Oracle XStream APIs to capture data changes and committed transactions from an Oracle primary database. The XStream APIs enable you to share data changes between Oracle and non-Oracle data sources, such as SAP® Replication Server®.

The Replication Agent for Oracle supports replication of:

- Supported datatypes specified in *Supported and Unsupported Datatypes* on page 4.
- Stored procedures and marker objects.
- Data in Oracle sequences.
- Data in a materialized view table.
- Compression; the level of compression support is dependent on the level of compression supported by Oracle XStream.
- Encryption; the level of encryption support is dependent on the level of encryption supported by Oracle XStream.

Replication Agent observes the same limitations for Oracle data as XStream APIs. For information about those limitations and Oracle XStream, see the Oracle documentation.

Supported Databases and JDBC Drivers

Check the supported Oracle database, and Java Database Connectivity (JDBC) driver versions required to use Replication Agent for Oracle with XStream APIs.

Database	Version
Oracle	11g Release 2 (patch 3 or later) –11.2.0.3 or later

Replication Agent for Oracle requires a JDBC 4.0-compliant driver for the Oracle primary data server.

Driver	Version
Oracle JDBC driver	11.2.0.3 or later

To use the standard JDBC 4.0 driver for Oracle database version 11.2.0.3:

- Download the `ojdbc6.jar` JDBC driver. See the Oracle documentation for instructions on how to download.
- Install the same version of the JDBC driver in the Oracle database environment in which Replication Agent resides.

Technical Prerequisites

Prerequisites for using the Replication Agent for Oracle with XStream APIs.

- The Replication Agent primary database user **pds_username** must have the XStream administrator privilege to:
 - Create or drop an XStream outbound server
 - Receive or send data changes from an XStream outbound server
 - Capture, synchronize, and apply data changes to an inbound server

To grant the XStream administrator privilege to the **pds_username**, issue:

```
BEGIN
DBMS_XSTREAM_AUTH.GRANT_ADMIN_PRIVILEGE( grantee =>
'pds_username' );
END;
/
```

See the Oracle documentation for the **DBS_XSTREAM_AUTH** package information.

- When you use XStream APIs to capture data changes, grant read/write access to the primary database archive logs to enable a non-Oracle Recovery Manager (RMAN) archive truncation method.

If you are using Oracle RMAN for archive truncation, or if the archive truncation method is disabled, read/write access to the archive logs is not required.

Depending on the configuration, Replication Agent may also access archived transactions logs (default) or may process only online transaction logs. For information about redo log and archive log files, see the *Replication Agent Primary Database Guide*.

- Before creating a Replication Agent instance, if the primary TNS service name is different from the **pds_database_name**, set the **pds_service_name** to the Oracle Transparent Network Substrate (TNS) service name found in the `$ORACLE_HOME/network/admin/tnsnames.ora` file.

For example, if `tnsnames.ora` contains these values:

```
ORCL112 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = huyharry) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl112.sybase.com)
    )
  )
```

Set the **pds_service_name** to `orcl112.sybase.com` instead of `orcl112` in the resource file.

- When you use XStream APIs to read from the transaction log, Replication Agent sets the Oracle Call Interface (OCI) server connection to Dedicated mode by default. If you change the OCI connection mode to Shared, the `ORA-26908` exception occurs. For mode descriptions and error exception details, see the Oracle documentation.
- When the **Oracle Database Vault** parameter is enabled at the primary database, you must grant the required privileges to the **pds_username** using the Oracle user login with the **DV_OWNER** role. Otherwise, you cannot execute the **pdb_xlog init** command to create, view, or remove the Replication Agent system objects.

To grant the required permissions to the **pds_username**:

1. Check whether **Oracle Database Vault** is enabled at the primary database by issuing:

```
SELECT * FROM V$OPTION WHERE PARAMETER = 'Oracle Database Vault';
```

A returned result of true indicates that the Oracle Database Vault is enabled, continue with next steps.

If the result is false, ignore remaining steps.

2. Execute **ra_admin init** to generate a SQL script called `prepare_pdb_with_vault_for_replication_init.sql` in the `<instance_directory>/scripts/prepare`, where *instance_directory* is the Replication Agent installation directory.
3. Run the `prepare_pdb_with_vault_for_replication_init.sql` script to grant the required privileges.

For more information about Oracle Database Vault, see the Oracle documentation.

Supported and Unsupported Datatypes

Datatypes that are supported and not supported by Replication Agent for Oracle 15.7.1 SP120 using XStream APIs.

Replicating Datatypes Out of Oracle

Oracle datatypes that are supported or not supported by Replication Agent for replicating out of Oracle using XStream APIs.

Table 1. Supported and Unsupported Oracle Datatypes

Supported	Unsupported
binary_double	bfile
binary_float	rowid
blob	ref(user-defined datatype)
clob	varray(user-defined object)
char	nested tables (user-defined object)
date	xmltype (such as binary xml)
interval day to second	
interval year to month	
number	
raw	
timestamp	
timestamp with [local] time zone	
timestamp with time zone	

Oracle-Supplied Datatype Limitations

Replication Agent cannot replicate some Oracle-supplied datatypes, because XStream APIs do not support them in row Logical Change Records (LCRs).

- “ANY” types (SYS.ANYTYPE, SYS.ANYDATASET)
- BFile
- ROWID
- User-defined datatypes, such as REF, including these object types:

- NESTED TABLE
- VARRAY
- XMLType object stored relationally or as a binary XML
- Media types (ORDSYS.ORDAudio, ORDSYS.ORDImage, ORDSYS.ORDImageSignature, ORDSYS.ORDVideo, ORDSYS.ORDDoc, SI_StillImage, SI_Color, SI_AverageColor, SI_ColorHistogram, SI_PositionalColor, SI_Texture, SI_FeatureList)
- Spatial types (MDSYS.SDO_GEOMETRY, SDO_TOPO_GEOMETRY, SDO_GEORASTER)
- Uniform Resource Identifier (URI) Types

These datatype restrictions are applicable to both ordinary (heap-organized) tables and index-organized tables. For descriptions about these tables, see the Oracle documentation.

Replication Agent for Oracle Limitations

The limitations for using Replication Agent for Oracle with XStream APIs.

- Because Replication Agent for Oracle does not support multiple XStream outbound servers, Multi-Path Replication™ (MPR) is not supported.
- Because XStream APIs return only committed transactions, Replication Agent for Oracle does not support replication of stored procedures in which autonomous transactions exist.
- Because the data definition language (DDL) LCR does not return the global temporary table details from an XStream outbound server, replication of temporary tables is not supported.
- Because XStream APIs does not support changes to a table created with the reference partition, Replication Agent for Oracle cannot replicate Data Manipulation Language (DML) statements on such tables.

DDL Replication Limitation

DDL replication is not certified with Replication Agent for Oracle 15.7.1 SP120 version when you use XStream APIs to read from Oracle database logs.

Unsupported DDL Configuration Parameters

DDL replication configuration parameters that are not supported.

Parameter	Description
pdb_auto_alter_repdefs	Determines whether Replication Agent automatically generates alter replication definition statements when processing changes in a marked table schema.

Parameter	Description
pdb_automark_tables	Determines whether Replication Agent automatically marks tables for replication during DDL replication.
lr_send_trunc_partition_ddl	Determines whether truncate partition commands are sent as DDL or DML to the replicate database.
ddl_password	Updates the log device repository in the RASD. Identifies the password for ddl_username .
ddl_username	The database user name included in Log Transfer Language (LTL) for replicating DDL commands to the standby database. This user must have permission to execute all replicated DDL commands at the standby database.

Unsupported DDL Commands

DDL replication command that is not supported.

pdb_setreddl – returns DDL replication status, and enables or disables replication for DDL statements.

See the *Replication Agent Reference Manual* for command and configuration parameters information.

Enhanced pdb_skip_op Command

The **pdb_skip_op** command now accepts a new parameter combination for Oracle XStream operations.

The format of the record locator is database-specific.

Syntax

You can specify this parameter combination to add, return, or remove XStream operations from a list of operations to skip:

```

pdb_skip_op [
{
add , { locator | {commitscn, scn, sequence, thread} }
|
remove, { all | locator | {commitscn, scn, sequence, thread} }
}
]
    
```

You can also use the new parameter combination in the **ra_helpop** command.

For other database parameter combinations, see the *Replication Agent Reference Manual*.

Parameters

- **add** – adds a specified ID to the list of identifiers of records to skip.
- **remove** – removes a specified ID from the list of identifiers to skip.
- **locator** – the locator keyword from the list of Log Transfer Manager (LTM) locators to identify the operations to skip.
- **commitscn** – (for Oracle) the system change number (SCN) keyword identifies a log record to skip.
- **scn** – (for Oracle) the SCN keyword identifies a specified log record to skip.
- **sequence** – (for Oracle) the sequence keyword identifies a specified log record to skip.
- **thread** – (for Oracle) the thread keyword of the redo log thread of the operation to skip.

Examples

- **Example 1** – adds an operation into skip list:

```
pdb_skip_op add, 0x0000.00425c72.0000, 0x0000.00425c72.0000, 1, 1
go
```

- **Example 2** – checks the current skip list:

```
pdb_skip_op
go
Skip List
-----
-----
SCN:0x0000.00425c72.0000, Thread:0x0001, Sequence:0x0000,
CommitSCN:0x0000.00425c72.0000
(1 row affected)
```

- **Example 3** – removes the skip operation from skip list:

```
pdb_skip_op remove,0x0000.00425c72.0000, 0x0000.00425c72.0000, 1,
1
go
```

- **Example 4** – checks the current skip list:

```
pdb_skip_op go
(0 rows affected)
```

Usage

- The **pdb_skip_op** command allows you to skip problem records, thereby avoiding having to reinitialize Replication Agent.
- The **pdb_skip_op** command is valid when Replication Agent is in the *Admin* or the *Replication Down* state.

Configuring the Replication Agent for Oracle to Use XStream APIs

Configure the Replication Agent for Oracle to use XStream APIs with a new **oraclexs** database type or a resource file.

1. Set the location of the Oracle XStream JDBC driver in the CLASSPATH environment variable
2. Create a Replication Agent instance with XStream APIs using the command line prompt, or, create a Replication Agent instance using a resource file:
 - Create and validate a resource file
 - Create the Replication Agent instance
3. Initialize a Replication Agent instance.
4. Mark tables, stored procedures, and Oracle sequences for replication.
5. Enable replication for marked tables, stored procedures, and sequences.
6. Truncate the transaction logs.
7. Back up Replication Agent objects in the primary database.
8. Deinitialize a Replication Agent instance.

Setting Up an Oracle XStream JDBC Driver

To use the Replication Agent for Oracle with XStream APIs, use the **setenv** function to set or modify ORACLE_HOME, LD_LIBRARY_PATH, and CLASSPATH environment variables.

Prerequisites

- Download and install the appropriate XStream JDBC driver from the Oracle Web site for your primary Oracle database environment in which Replication Agent resides.
- Make sure your Java Runtime Environment (JRE) is compatible with your Oracle database for using XStream APIs. For example, if you have installed 64-bit Oracle, use the 64-bit JRE 1.7.x. See the Oracle documentation.

By default, Replication Agent uses 64bit JRE when creating an instance with the XStream APIs Log Reader component.

Task

1. Set the ORACLE_HOME environment variable to the Oracle installation directory for your platform:

```
setenv $ORACLE_HOME <install_dir>/oracle/<version>/
```

where *install_dir* is the installation directory and *version* is the version number of the Oracle database.

The Oracle installation directory might be different, based on the database version installed on your platform. Set the `ORACLE_HOME` variable before setting the `LD_LIBRARY_PATH` and `CLASSPATH` environment variables.

2. Add the path of the Oracle XStream JDBC driver to the `CLASSPATH` environment variable for your platform.

Add only one version of a JDBC driver in the `CLASSPATH`; otherwise, Replication Agent will have problems connecting to the primary database.

On Linux:

```
setenv LD_LIBRARY_PATH $ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

```
setenv CLASSPATH $ORACLE_HOME/jdbc/lib/ojdbc6.jar:$ORACLE_HOME/
rdbs/jlib/xstreams.jar:$CLASSPATH
```

On HP-UX64:

```
setenv SHLIB_PATH $ORACLE_HOME/lib:$SHLIB_PATH
```

```
setenv CLASSPATH $ORACLE_HOME/jdbc/lib/ojdbc6.jar:$ORACLE_HOME/
rdbs/jlib/xstreams.jar:$CLASSPATH
```

On AIX 64:

```
setenv LIBPATH $ORACLE_HOME/lib:$LIBPATH
```

```
setenv CLASSPATH $ORACLE_HOME/jdbc/lib/ojdbc6.jar:$ORACLE_HOME/
rdbs/jlib/xstreams.jar:$CLASSPATH
```

On SUNOS-i86pc:

```
setenv LD_LIBRARY_PATH $ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

```
setenv CLASSPATH $ORACLE_HOME/jdbc/lib/ojdbc6.jar:$ORACLE_HOME/
rdbs/jlib/xstreams.jar:$CLASSPATH
```

On SUNOS-64:

```
setenv LD_LIBRARY_PATH_64 $ORACLE_HOME/lib:$LD_LIBRARY_PATH_64
```

```
setenv CLASSPATH $ORACLE_HOME/jdbc/lib/ojdbc6.jar:$ORACLE_HOME/
rdbs/jlib/xstreams.jar:$CLASSPATH
```

where:

- *Oracle_Home* – is the Oracle database installation directory.
- *path_name* – is the library path where you installed the JDBC driver:
 - *LD_LIBRARY_PATH* on UNIX
 - *SHLIB_PATH* on HP-UX64
 - *LIBPATH* on AIX
- *driver* – is the name of the JDBC driver; for Oracle 11.2.0.3, `ojdbc6.jar`. The `xstreams.jar` is the name of the XStream library path.

3. (On Windows) Set Oracle environment variables in any of these ways:

<p>Use the Control Panel</p>	<ul style="list-style-type: none"> • Select Start > Settings > Control Panel > System > Advanced System Settings > Environment Variables. • Click New in the User variables pane. • In the New User Variable dialog, enter: <ul style="list-style-type: none"> • Variable name – is the environment variable that you want to set. For example, ORACLE_HOME, PATH, and CLASSPATH. • Variable value – is the directory path that you want to set for the environment variable. For example, c:\oracle\version (installation directory), C:\ORACLE_HOME\bin (the library path), and C:\ORACLE_HOME\jdbc\lib\ojdbc6.jar (the JDBC driver path).
<p>Edit the existing variables</p>	<ul style="list-style-type: none"> • Select the PATH and CLASSPATH environment variables in the System variables pane, and click Edit. • In the Edit System Variable dialog, using the semicolon (;) path separator, enter the variable value for: <ul style="list-style-type: none"> • Path – %ORACLE_HOME%\bin;%ORACLE_HOME%\LIB;%path% • CLASSPATH – %ORACLE_HOME%\jdbc\lib\ojdbc6.jar;%ORACLE_HOME%\rdbms\jlib\xstreams.jar;%CLASSPATH%
<p>In the command line prompt</p>	<p>Enter:</p> <pre>set CLASSPATH=ORACLE_HOME\jdbc\lib\ojdbc6.jar;ORACLE_HOME\rdbms\jlib\xstreams.jar</pre>

Creating a Replication Agent Instance

Create a Replication Agent instance, which uses XStream APIs to capture replicated transactions from an Oracle primary database.

At any time after the Replication Agent software is installed, invoke **ra_admin** with the **-c** option:

```
ra_admin -c new_inst -p port_num {-t database|-f old_inst} [-uid ra_username] [-pwd ra_password]
```

For parameter descriptions, see the **ra_admin** command in the *Replication Agent Administration Guide*.

In this example, you create a Replication Agent instance named **raoxs**, which listens to port 8888 and database type is **oraclexs** for Oracle XStream:

```
ra_admin -c raoxs -t oraclexs -p 8888
```

The **oraclexs** parameter initializes the Replication Agent Log Reader component to use XStream APIs to capture committed transactions.

Creating a Replication Agent Instance Using a Resource File

Create a Replication Agent instance using the **oraclexs** resource file template for Oracle XStream.

Invoke **ra_admin** using the command line parameters that create and validate a Replication Agent instance using a resource file:

```
ra_admin {-vr res_file | -r res_file}
```

where:

- **-vr res_file**
validates the specified resource file (*res_file*), without creating a Replication Agent instance or making any change in the environment.
- **-r res_file**
creates a Replication Agent instance, based on the contents of the specified resource file (*res_file*).

A resource file is an ASCII text file that contains configuration information for the Replication Agent instance to be created by **ra_admin**.

In addition to creating a Replication Agent instance, the **ra_admin** parameters in the resource file allows you to:

- Create the user login in the primary data server, and grant all required permissions.
- Start the new instance after it is created.
- Initialize the new instance after it starts.

To create a Replication Agent instance with XStream APIs, set the **instance_type** configuration parameter in the resource file as `instance_type=oraclexs`.

Note: When you create a Replication Agent instance with a resource file, the **pds_username** and **pds_password** configuration parameter values cannot contain single or double quotes if the **create_pds_username** parameter is set to yes.

Creating and Validating a Replication Agent Resource File

Create a resource file using the **oraclexs** template and validate the file by issuing the **ra_admin** command with the **-vr** option.

Prerequisites

Locate your resource file templates for Oracle database in the `init` subdirectory of the Replication Agent installation directory. For example:

```
C:\sybase\RAX-15_5\init\oraclexs.rs
```

This resource file template contains comments that describe each configuration parameter and its value.

Note: It is recommended that you to validate each resource file before you create a Replication Agent instance using that resource file.

Task

1. Copy the resource file template to another file that you edit to create the new resource file.

For example:

```
cp oraclexs.rs raoxs.rs
```

where `raoxs.rs` is the name of the new resource file you want to create.

2. Use a text editor to edit the resource file copy that you created.
3. Validate your resource file by executing:

```
ra_admin -vr raoxs.rs
```

where `raoxs.rs` is the name of the resource file you want to validate.

Validation results are returned as one of:

- Response-file processing completed.
- Response-file processing completed with errors.

If the validation is successful, skip step 4, and use the resource file to create a Replication Agent instance.

4. To correct validation errors:
 - a) Review the error messages to determine the cause of the failure.
 - b) Edit the resource file to correct the appropriate values.
 - c) Re-run **ra_admin -vr** specifying the name of the resource file.

Repeat this step until the resource file is successfully validated.

Creating a Replication Agent Instance with a Resource File

Create a Replication Agent for Oracle instance and specify the resource file at the same time.

When you run **ra_admin** with the **-r** option, the utility first validates the specified resource file except:

- If the Replication Agent for Oracle primary database user login does not exist in the primary data server, the utility creates it, if specified in the resource file (`create_pds_username=yes`).
- If the user login exists in the primary data server but does not have all the required privileges, set **create** to yes, to have the utility grant all required permissions.
- If the resource file specifies to initialize new Replication Agent instance (`initialize_instance=yes`), then:
 - The Replication Agent primary database user login must either exist in the primary data server, or be created by **ra_admin** (`create_pds_username=yes`).
 - The resource file must specify the instance type for Oracle XStream (`instance_type=oraclexs`).

- The resource file must specify to start the Replication Agent instance (`start_instance=yes`).

Otherwise, the utility returns an error message and does not create the instance.

Note: We recommend you to validate a new resource file before you create a Replication Agent instance using the new resource file.

Execute **ra_admin**, specifying the **-r** option and the name of the resource file:

```
ra_admin -r c:/raoxs.rs
```

where `raoxs.rs` is the name of the resource file.

Results are returned as one of:

- Response-file processing completed.
- Response-file processing completed with errors.

If the instance creation is successful, the `oraclexs.cfg` configuration file is created in `$SYBASE/RAX/15_5/config` directory. You can begin using the new Replication Agent instance.

The configuration file now includes the XStream specific LogReader and LogAdmin
'`ra_log_reader=com.sybase.ra.lr.oracle.xstream.RAOXSLogReader`'
and
'`ra_log_admin=com.sybase.ra.la.oracle.xstream.RAOXSLogAdmin`'.

Initializing a Replication Agent Instance to Use Oracle XStream APIs

Initialize an instance of Replication Agent, and create the objects it needs in the primary database.

Prerequisites

The Replication Agent instance must be running, and connectivity to the primary database must be established.

Task

1. Log in to the Replication Agent instance with the administrator login.
2. Run **ra_admin init**, where **init** is the keyword for creating Replication Agent system objects in the primary database.

If the XStream administrator privilege is granted to the **pds_username**, an XStream outbound server is created during initialization.

If the initialization is successful, the script is stored in a file named `partinit.sql` in the `scripts/xlog/installed` directory.

If the initialization does not succeed, see *Replication Agent Administration Guide* for troubleshooting information.

Deinitializing a Replication Agent Instance

Deinitialize an instance of Replication Agent, if that instance is no longer required, and remove its objects from the primary database.

Prerequisites

The Replication Agent instance must be running in the *Admin* state to remove its objects from the primary database and to deinitialize Replication Agent.

Task

1. Log in to the Replication Agent instance with the administrator login.
2. Verify that the Replication Agent objects exist in the primary database:

```
ra_admin
```

If there are no Replication Agent objects in the primary database, the **ra_admin** command returns no information, and you do not need to complete this task.

If objects exist for this Replication Agent instance, the **ra_admin** command returns a list of the names of the objects.

3. Disable replication of all marked tables in the primary database:

```
pdb_setreptable all, disable
```
4. Disable replication for all marked procedures in the primary database:

```
pdb_setrepproc all, disable
```
5. Unmark all marked tables and stored procedures in the primary database:

```
pdb_setreptable all, unmark
```



```
pdb_setrepproc all, unmark
```
6. Remove the Replication Agent objects:

```
ra_admin deinit
```

After you invoke the **ra_admin** command with the **deinit** keyword, Replication Agent generates a script that removes the objects from the primary database, and deinitializes Replication Agent.

If the deinitialization was successful, the script is stored in a file named `cleanup_pdb_for_replication_deinit.sql` in the `RAX-15_5\inst_name\scripts\cleanup` directory. Execute this script to remove Replication Agent instance objects.

Upgrading the Replication Agent for Oracle from LogMiner to XStream

To upgrade the Replication Agent for Oracle Log Reader component from LogMiner to XStream APIs, use the **ra_admin** command with the **-u** option.

Prerequisites

- Make sure the primary database version is Oracle11.2.0.3 or later.
- Initialize the Replication Agent.
- Set the Oracle library path and the XStream JDBC driver in the CLASSPATH environment variable for your platform. See *Setting Up the Oracle XStream JDBC Driver* on page 8.
- Shut down the Replication Agent instance and make sure no database operations are running to avoid risking data loss due to missed replication.

Note: You cannot downgrade to LogMiner once you have upgraded the Replication Agent Log Reader to use XStream APIs.

Task

1. To use maximum-strength password encryption, install Java Cryptography Extension (JCE) policy files and reset the system administrator account.
See Installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 7 on page 21.
2. Set the SYBASE environment variable by changing the SYBASE directory in which Replication Agent is installed.
 - For C Shell: `source SYBASE.csh`
 - For Bourne or Korn shell: `. SYBASE.sh`
 - For Windows, run: `SYBASE.bat`
3. Change to the Replication Agent bin directory using the command line prompt:
 - On UNIX:

```
cd $SYBASE/RAX-15_5/bin
```
 - On Windows:

```
cd %SYBASE%\RAX-15_5\bin
```
4. Upgrade the Replication Agent instances to use Oracle XStream APIs.

<p>Upgrading instances within the same product installation</p>	<ul style="list-style-type: none"> To upgrade a specific Replication Agent instance in the Replication Agent installation directory, at the command prompt, run: On UNIX: <pre>./ra_admin.sh -u instance=<instance_name> -t oraclexs</pre> On Windows: <pre>ra_admin.bat -u instance=<instance_name> -t oraclexs</pre> To upgrade all Replication Agent instances in the Replication Agent installation directory, at the command prompt, run: <pre>./ra_admin.sh -u all -t oraclexs</pre> <pre>ra_admin.bat -u all -t oraclexs</pre> <p>Note: Because the current version upgrade is not supported, you cannot upgrade the Replication Agent for Oracle 15.7.1 SP120 Log Reader component from LogMiner to use XStream APIs.</p>
<p>Upgrading instances in a separate product installation</p>	<ul style="list-style-type: none"> To upgrade a specific Replication Agent instance in the specified source instance directory from the current product installation directory, run: On UNIX: <pre>./ra_admin.sh -u <source_instance_directory> -t oraclexs</pre> On Windows: <pre>ra_admin.bat -u <source_instance_directory> -t oraclexs</pre> To upgrade all Replication Agent instances in the specified source installation directory from the current product installation directory, at the command prompt, run: <pre>./ra_admin.sh -u <source_installation_directory></pre> <pre>ra_admin.bat -u <source_installation_directory></pre> <p>If the Oracle primary database version is earlier than 11.2.0.3, the upgrade process stops and shows: "The Replication Agent upgrade is not supported on <instance_name> for database version <xx.xx.xx.xx>. The minimum database version supported to upgrade Replication Agent from LogMiner to XStream APIs is:11.2.0.3."</p>

In the commands above:

- all* – upgrades all instances of Replication Agent within the current product installation directory except the Replication Agent 15.7.1 SP120 instance with the LogMiner

component. The configuration files are backed up before the upgrade. If an error occurs, the upgrade is rolled back.

- *instance_name* – is the name of the Replication Agent instance to be upgraded. When you enter *instance_name*, the upgrade is performed directly on the specified instance. The configuration file is backed up before the upgrade for use in error recovery, if required. If an error occurs, the upgrade is rolled back.
- *source_installation_directory* – is the Replication Agent source installation directory. When you enter the *source_installation_directory*, all instances of Replication Agent in the specified source installation directory are upgraded from the current product installation directory provided the earlier version of the Replication Agent instance uses LogMiner component to read from the transaction logs. The instance is first copied to and then updated within the current product installation directory
- *oraclexs* – is the Oracle XStream database type. When you enter `-t oraclexs`, the Log Reader component is upgraded from LogMiner to Oracle XStream APIs.

5. Start the Replication Agent instance.

On UNIX:

```
$SYBASE/RAX-15_5/<instance>/RUN_<instance>
```

On Windows:

```
%SYBASE%\RAX-15_5\<instance>\RUN_<instance>
```

where *instance* is the Replication Agent instance name.

6. Log in to the Replication Agent instance and migrate the metadata by running:

```
ra_migrate
```

This command initiates the upgrade process, finalizes the upgrade of an instance from an earlier version, and prevents a subsequent downgrade to the earlier version.

During upgrade, Replication Agent creates an XStream outbound server with the current start system change number (SCN) to read from transactions logs.

7. Once the Replication Agent is upgraded, set the Oracle environment variables for XStream APIs.

8. Execute the Replication Server System Database (RSSD) **rs_zeroltm** command at the primary database connection to reset the LTM locator value to zero.

Run **rs_zeroltm** before resuming replication, to prevent Replication Server from requesting a log starting point that occurs earlier in the log than the location established by the **ra_migrate** option. The Replication Server LTM locator value for the primary connection must be zeroed.

For example:

```
rs_zeroltm data_server, database
```

9. Resume replication.

```
resume
```

Before resuming replication, wait for the upgrade process to complete. Otherwise, data inserted during upgrade is not replicated.

Enhancements in Replication Agent for Microsoft SQL Server

Several enhancements are available in Replication Agent for Microsoft SQL Server in the 15.7.1 SP120 version.

Replicate Microsoft SQL Server Large Object Datatypes into an SAP HANA Database

Replication Agent for Microsoft SQL Server 15.7.1 SP120 now supports replication of `varchar(max)`, `nvarchar(max)`, and `varbinary(max)` large object (LOB) datatypes into an SAP HANA® replicate database.

In earlier versions, `varchar(max)`, `nvarchar(max)`, and `varbinary(max)` LOB datatypes were considered as a partial update, even for a full LOB update. Replication of partial updates to LOB datatypes was supported only in the Microsoft SQL Server to Microsoft SQL Server (homogeneous) replication environment.

With Replication Agent for Microsoft SQL Server 15.7.1 SP120, you can use the **`lr_partial_updates_allowed`** configuration parameter to determine whether replicate databases support partial updates to LOB datatypes.

By default, **`lr_partial_updates_allowed`** is true.

When the **`lr_partial_updates_allowed`** is true, Replication Agent sends changes to LOB datatype columns as partial updates to Replication Server. Replication Server sends partial updates to the replicate databases that have native support for partial LOB updates, such as Microsoft SQL Server.

If the replicate database do not support partial updates to LOB datatypes, set the **`lr_partial_updates_allowed`** to false.

When the **`lr_partial_updates_allowed`** is false, Replication Agent converts partial LOB updates to a full update, which is consumed in heterogeneous replication environments. If partial LOB updates are detected, an error exception occurs.

To replicate LOB datatypes from the Microsoft SQL Server to SAP HANA replicate database, set the **`lr_partial_updates_allowed`** parameter to false.

For Microsoft SQL Server LOB datatype mappings and limitations, see the *Replication Server Administration Guide: Volume 1*.

Enhanced server_xlog Command

The **server_xlog** command is now replaced with **server_admin** to be consistent with other Replication Agent commands.

Migrating Replication Agent for Microsoft SQL Server When Upgrading SQL Server Database

Migrate Replication Agent for Microsoft SQL Server when you upgrade SQL Server database from 2008 to 2008R2 version.

1. To prevent loss of any replicated data, deny users—other than existing Replication Agent **pds_username** users—access to the primary database.
See the Microsoft SQL Server documentation.
2. Log in to the Replication Agent for Microsoft SQL Server 15.7.1 SP120 instance, verify that it is in the *Replicating* state, and allow replication to finish.

To verify that replication has completed:

- a) Periodically issue the **ra_statistics** command, until all of these statistics are 0 (zero):
 - Input queue size
 - Output queue size
 - b) Note the `Last QID Sent` from the last set of statistics.
 - c) Issue the **ra_locator update** command so that Replication Agent retrieves the truncation point from Replication Server.
 - d) Issue the **ra_locator** command again and compare the displayed locator with that of the `Last QID Sent`. If they are different, wait and repeat this step.
 - e) Quiesce the Replication Agent instance by issuing the **quiesce** command.
 - f) Shut down the Replication instance by issuing the **shutdown** command.
3. Upgrade Microsoft SQL Server database from 2008 to 2008R2 version, following the steps in the Microsoft SQL Server documentation.
 4. Start the primary database in single-user mode:


```
"C:\Program Files\Microsoft SQL Server\MSSQL<version>\MSSQL\Binn\sqlservr.exe" -m -s instanceName
```

where:

 - *version* – is the version number of the database.
 - *instanceName* – is the name of the Microsoft SQL Server instance.
 5. Start the Replication Agent instance in one of these ways:
 - Invoke the RUN script for the instance that you want to start.
 - Invoke **ra**, and specify the instance that you want to start.

- If Replication Agent is installed on a Windows system, start a Windows service for the instance that you want to start.

The RUN script and the **ra** utilities are batch files on Microsoft Windows.

See *Replication Agent Start-Up* in the *Replication Server Administration Guide* to start the instance on your platform.

6. Initialize Replication Agent system database objects by issuing **server_admin init**.
7. Restart the primary Microsoft SQL Server database in multiuser mode:
 - a) From Windows Control Panel, go to **Administrative Tools > Services**, and find the service named SQL Server (SERVER).

where *SERVER* is the name of your Microsoft SQL Server data server.
 - b) Start this service.
8. Migrate the Replication Agent metadata by issuing **ra_migrate**.
9. In the Replication Agent instance, resume replication by issuing **resume**.
10. Allow all users to access the primary database.

Replication Agent Security Enhancements

Replication Agent 15.7.1. SP120 includes several security enhancements.

Support for Maximum-Strength Password Encryption

The Replication Agent 15.7.1 SP120 version does not include Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files for maximum-strength (256-bit) password encryption.

By default, Replication Agent 15.7.1 SP120 now uses 128-bit password encryption string. To use the maximum-strength password encryption string, you must install JCE policy files and reset the system administrator account. Otherwise, you see an error message when you:

- Upgrade an earlier version of Replication Agent 15.7.1 SP100 to 15.7.1 SP120 version using the **ra_admin -u instance=<instance_name>** command.

Could not decrypt the encrypted passwords because the encryption algorithm has been changed.

Please verify if Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files is installed or reset administrator account.

- Run the **ra_downgrade** command to downgrade the Replication Agent 15.7.1 SP120 to 15.7.1 SP100 version.

Downgrade to previous version may require Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

For upgrade and downgrade instructions, see *Replication Agent Primary Database Guide*.

Installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 7

To use 256-bit password encryption, install JCE Unlimited Strength Jurisdiction policy files in the Replication Agent `security` directory.

1. Shut down the Replication Agent instance.
2. Download the JCE Unlimited Strength Jurisdiction Policy Files 7 from the Oracle Web site.
3. Extract the `UnlimitedJCEPolicyJDK7.zip` and copy the content of the Zip file in the Replication Agent installation directory.
 - On Windows – `%SYBASE%\RAX-15_5\JRE7\lib\security`
 - On UNIX – `$(SYBASE)/RAX-15_5/JRE7/lib/security`
4. Delete the existing password encryption string:

```
ra_admin -k <instance>
```

5. To re-generate the new maximum-strength password encryption string, reset the system administrator account:

```
ra_admin -reset <instance>
```

See *Resetting the Replication Agent Administrator Password* in the *Replication Agent Administration Guide*.

6. Restart the Replication Agent instance.
7. Resume replication:

```
resume
```

SAP JVM Support

The Replication Agent 15.7.1 SP120 installer includes the SAP JVM.

For more information about the SAP JVM, see the *SAP Help Portal*.

You can modify the Open Client/Server™ trusted roots file to include trusted CA certificates.

The list of known and trusted CA certificates is maintained in the trusted roots file:

- On UNIX – `$(SYBASE)/RAX-15_5/config/trusted.txt`
- On Windows – `%SYBASE%\RAX-15_5\config\trusted.txt`

The system administrator can use a standard ASCII-text editor to add and delete CA certificates.

By default, Replication Agent recognizes these third-party CA certificates:

- Baltimore Cyber Trust

New Features in Replication Server Options 15.7.1 SP120

- Entrust
- Equifax SCA
- GTE Global
- SAP Passport
- Trustcenter
- Telekom
- thawte
- VeriSign
- Workplace ca