

Oracle Service Cloud

Data Volume Management

ORACLE WHITE PAPER / FEBRUARY 25, 2019

Table of Contents

Introduction	4
Intended Audience	4
Oracle service Cloud Platform – Data Volume Management.....	4
Why data volume management is crucial	4
DB tables of interest	4
Monitoring Database Usage	5
Oracle Cloud Portal	5
Configuration Assistant.....	5
Data Volume Management	7
Data Lifecycle Management (DLM)	7
Using configuration verbs to define purge policies	9
Bulk Delete API	10
Using custom objects for logging.....	11
Why are large database reductions not being seen?.....	12
High volume tables vs. course of action	12
File Attachment Server (FAS) volume management.....	14
Best Practices.....	14
Conclusion	15

Additional Resources.....15

INTRODUCTION

This whitepaper provides an insight into various data volume management capabilities offered by Oracle Service Cloud. This allows customers to efficiently manage data storage on their site and meet their organizational compliance policies.

INTENDED AUDIENCE

This whitepaper is a useful read for all Oracle Service Cloud administrators, partners, technical consultants and Customer Success Managers (CSMs).

ORACLE SERVICE CLOUD PLATFORM – DATA VOLUME MANAGEMENT

Oracle Service Cloud enables organizations to deliver great customer experiences through various channels including email, co-browse, telephony, live chat and other social channels.

Why data volume management is crucial

Oracle Service Cloud database is a transactional system storing all interactions processed from customers, agents and end-users. The backend database is designed and optimized to process individual transactions efficiently and is not intended to be used for long-term trending and reporting needs. Hence it is critical to keep the database size in check for the product to be highly performant and agile. An optimized and healthy database offers various benefits like easier and faster updates, faster report execution, faster database cloning, faster recovery from catastrophic failure and an overall improved performance. In short, efficient data management in Oracle Service Cloud helps you provide a greater customer experience that your customers deserve.

In most cases, the primary reason for the excessive database growth is due to retention of historical data in the database, which will in turn have negative consequences on your Service Cloud site's performance. It is critical that data management policies be set up to clean historical data and keep a consistent database size either by deleting them from the database or by moving it to a data warehouse for longer-term reporting and trending needs. Some downsides of historical data retention in Service Cloud include potentially significant performance impact for both customers and agents, increased operational costs, reports taking longer time to run, failures in updates because replication is falling behind, utilities taking longer time to run and database clones taking longer time to complete.

It is important to note that the larger the database size, the more time it takes to complete all the crucial data operations. Hence, an appropriate set of data volume management policies are key to maximize Oracle Service Cloud success.

DB tables of interest

We often notice some of the following tables could grow large very quickly:

- incidents
- threads
- phrases

- transactions
- user_trans
- co_trans
- inc_performance
- papi_meters/papi_method_meters
- clickstreams
- contacts
- archived_incidents
- custom objects
- PS\$log

Oracle Service Cloud provides tools and mechanisms to help you establish a data volume management strategy that we will cover in detail in the subsequent sections.

MONITORING DATABASE USAGE

The first and foremost step for efficient data volume management is to know what your current database usage is. “What makes up the size of my site and database” is one of the most frequently asked questions by the Service Cloud customers. Following data monitoring tools will help you to pro-actively monitor the database storage size and usage of your Oracle Service Cloud site:

- Oracle Cloud Portal
- Configuration Assistant

Oracle Cloud portal can be used to view the overall database usage of your Service Cloud site while Configuration Assistant can be used to view the table-level database usage.

Oracle Cloud Portal

Oracle Cloud Portal (cloud.oracle.com) houses self-service tools that Oracle Service Cloud customers can use to manage their sites and instances. You can use Cloud Portal to view and monitor the resource utilization and database usage for your Oracle Service Cloud site, among other things.

To access the cloud portal, launch <https://cloud.oracle.com/home>

Additional information on using the Oracle Cloud Portal can be found at:

http://docs.oracle.com/cloud/latest/trial_paid_subscriptions/index.html

Configuration Assistant

Oracle Service Cloud Configuration Assistant is a self-service tool, found within the MyServices area of the Cloud Portal, that allows you to complete administrative tasks for Oracle Service Cloud and also helps you to monitor storage usage data. The storage statistics seen in the configuration assistant will help you determine whether the database storage limits for a service are under-utilized or over-utilized. You can also use the statistics to manage your database storage by deleting, purging or archiving data that exceeds the storage limits.

Link to the Configuration Assistant is located on the Oracle Cloud My Services dashboard. You will receive the login credentials for My Services dashboard in a welcome email from Oracle Cloud. The Configuration Assistant home page is displayed based on the products or services you have purchased, i.e., Oracle Service Cloud.

To view the current usage statistics, click Statistics from the Configuration Assistant home page. The record count and storage graph displays the average record count and data storage statistics for the top 10 tables (by data size) for the latest available week, by data size.

The Data storage table displays the average record count and data storage statistics for all tables, for the latest available week, sorted by total storage. The Data Storage table appears as follows:

Data Storage ?

Menu ▾ View ▾ Format ▾ Freeze Wrap

Table Name	Record Count(K)	Storage(KB)	Index Size(KB)	Percent
Phrases	2,673,789	133,738,409	70,396,580	14.47
Contacts	129,664	15,288,240	63,606,424	5.59
Incident Performance	239,342	18,724,628	42,872,572	4.36
Opportunity Phrases	478,477	22,569,550	32,148,226	3.87
Opportunities	36,103	14,012,121	31,554,666	3.23
Marketing Activities	120,775	7,730,809	34,322,549	2.98
Incidents	45,331	10,387,493	21,838,575	2.28
Products to Quotes	86,873	10,957,600	3,071,246	0.99
Archived Incidents	42,957	6,689,910	6,515,851	0.93
Deleted Records	65,411	5,712,642	4,322,230	0.71

Record Count: 404

Figure 1. Data Storage table in Configuration Assistant

More details on the Configuration Assistant can be found at:

https://docs.oracle.com/en/cloud/saas/service/18a/facca/managing-ssl-certificates.html#c>About_monitoring_data_and_usage_statistics_ed7E90C1

DATA VOLUME MANAGEMENT

Key to efficient database volume management is to pro-actively monitor database usage and set data lifecycle policies to continually purge older data. Since each organization's compliance needs may be different, you can determine and set the data retention and purge policies based on your organization's compliance requirements, more aggressive the better.

Once you have identified the list of tables that make up the bulk of size of your database using the data monitoring tools, use one or more of the tools and capabilities offered within Oracle Service Cloud to manage the size of your database and to avoid being charged for extra storage:

- Data Lifecycle Management (DLM)
- Configuration settings
- Bulk Delete API

Data Lifecycle Management (DLM)

Data Lifecycle Management (DLM) provides an easy-to-use component with built-in policies to archive (only supported on Incidents) and purge unused data automatically and periodically. It is a self-service offering that enables customers to configure and host data lifecycle policies for objects. DLM currently operates on the following objects: Incidents, Contacts, Transactions and Custom Objects. You can define lifecycle policies on these objects to purge/archive unused records based on filter conditions you specify.

Here is an image of the DLM page displayed in BUI:

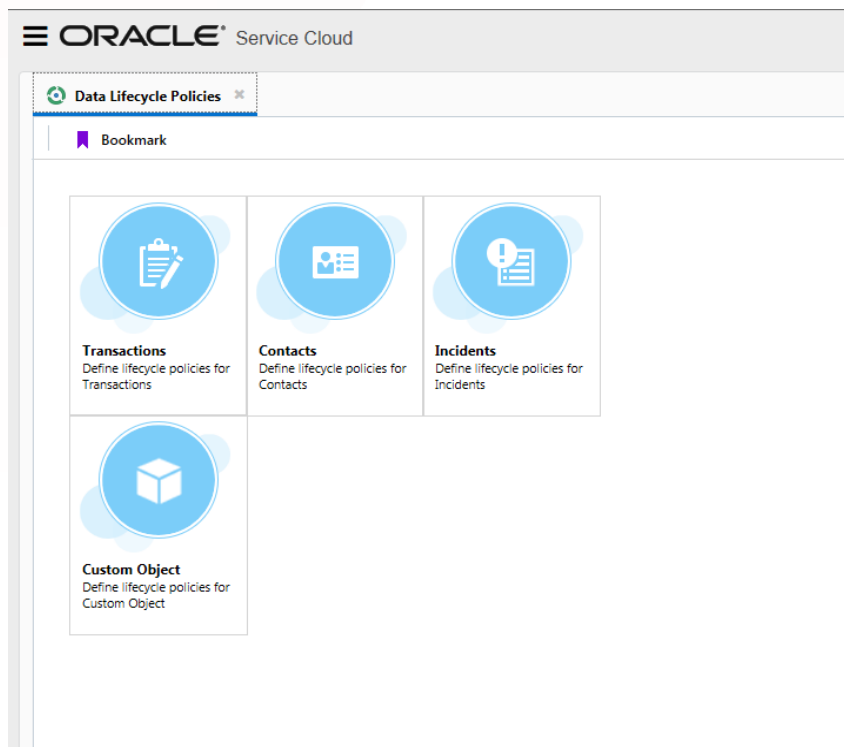


Figure 2. Data Lifecycle Management landing page

There are two types of DLM policies: Standard and Custom.

STANDARD DLM POLICIES

These are pre-defined policies and exist on all sites, may be enabled or disabled. Actions (i.e. purge/archive specific actions) are not editable on standard policies. Pre-defined filters cannot be deleted, however can be edited in some cases.

These are standard DLM policies available in the 18B or later releases:

- Transactions
 - Purge Incident.Transactions.TransEdit Older than 6 months
 - Purge Incident.Transactions.TransStatus Older than 6 months
 - Purge Incident.Transactions.TransQueue Older than 6 months
 - Purge Contact.Transactions.TransEdit Older than 6 months

Here are the definitions for different transaction types (used in the above policies):

- TransEdit is created during any update operation on any object type
- TransStatus is created when incident status is changed
- TransQueue is created when incident queue is changed
- Incidents
 - Purge Closed Incidents
 - For new 18B or later sites: The site level policy is disabled and the interface specific policies do not exist. If needed, create interface specific custom policies manually.
 - For sites upgrading to 18B or later releases: The site level policy is disabled. The interface specific custom policies are created based on the configuration verb PURGE_DELETE_INCIDENTS value. If the config value > 0, the policy is enabled with number of days as provided in the config verb. If not, it is disabled.
 - Archive Closed Incidents
 - For new 18B or later sites: The site level policy is enabled by default and the number of days is set to 365.
 - For sites upgrading to 18B or later releases: The site level policy is disabled. The interface specific custom policies are created based on the configuration verb ARCHIVE_INCIDENTS value. If the config value > 0, the policy is enabled with number of days as provided in the config verb. If not, it is disabled.

CUSTOM DLM POLICIES

You can create, update and delete custom policies on incidents, contacts and custom objects. Custom policies vary in number among sites. An example custom policy on Incidents would be to archive all solved incidents older than 9 months based on a custom attribute on Incident object. Currently all custom attributes and limited standard fields are available for defining custom DLM policies. Custom fields are not supported yet.

The below image shows an example of a custom DLM policy that archives all solved incidents older than 9 months on a specific site interface.

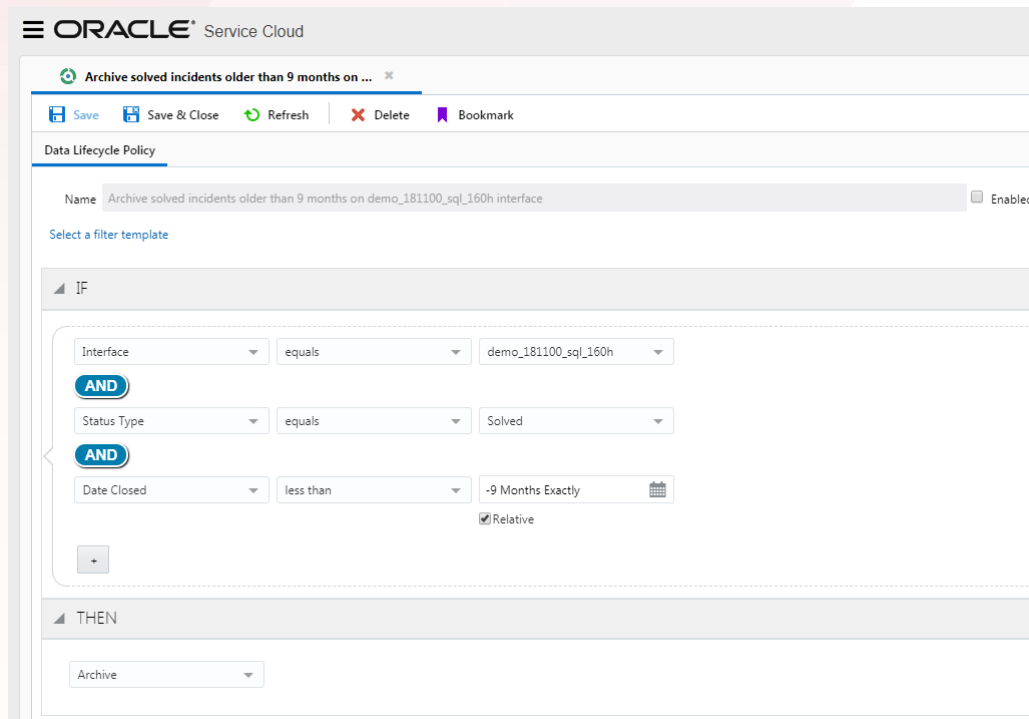


Figure 3. A Custom DLM Policy Example

SUMMARY

A data lifecycle policy, once enabled, will continue to exist and run indefinitely. Data, once purged, is irretrievable. So please be careful and thoughtful before enabling such a policy.

Here are some useful resources on DLM for further details:

https://cx.rightnow.com/app/answers/detail/a_id/10013

https://cx.rightnow.com/app/answers/detail/a_id/10538

http://documentation.custhelp.com/euf/assets/devdocs/buiadmin/topicrefs/c_bui_Data_lifecycle_management.html

Using configuration verbs to define purge policies

[Configuration settings](#) within the Oracle Service Cloud application allow you to configure the data retention period for a certain set of tables in your database. Based on your organization's data retention requirements and compliance regulations, set the values for these configuration settings accordingly.

In order to set the configuration settings, select Configuration > Site Configuration > Configuration Settings > and search by Key from the Navigation area in Agent console

PURGE CONFIGURATION SETTINGS

Purge Configuration Settings

CONFIGURATION SETTING	DEFINITION	TABLE ADDRESSED
PURGE_ARCHIVED_INCIDENTS	Defines the number of days after which archived incidents will be deleted from the archive. Set this value to 0 to prevent archived incidents from being deleted. The default is 1825.	archived_incidents +reduction in FAS storage by X MB/GB
PURGE_DELETE_INC_PERFORMANCE	Defines the number of days after which incident performance statistics are removed from the database. Set this value to 0 to prevent statistics from being removed from the database. The default is 0.	inc_performance
PURGE_DELETE_USER_TRANS	Defines the number of days after which login/logout sessions are removed from the user_trans table (relative to the logout time). Set this value to 0 to prevent login/logout sessions from being removed from the database. The default is 0.	user_trans
PURGE_DELETE_INCIDENT_PHRASES	Specifies the number of days after which solved incidents cease to be fully searchable. Solved incidents whose closed date is older than PURGE_DELETE_INCIDENT_PHRASES days will only support searching on keywords in the subject. Set this value to 0 or to a value greater than ARCHIVE_INCIDENTS to disable this feature. Note: Sites with less than 1 million rows in the incidents table are not affected. The default is 366.	phrases
PURGE_SESSION_TRACKING	Specifies the number of days after which customer session tracking data will be deleted. The default is 30.	clickstreams

There are additional configuration settings available to set the purge policies for other objects. Please find more details here: https://cx.rightnow.com/app/answers/detail/a_id/2362

Bulk Delete API

Bulk Delete API allows you to delete a large number of records from the database efficiently with minimal impact on the resources and site operations. Bulk delete API is available in all Connect bindings: Connect REST, Connect web services for SOAP and Connect PHP API.

SUPPORTED OBJECTS

Bulk Delete API can be to delete unused or unrequired records from the following objects:

- Incidents
- Opportunities
- Contacts
- Accounts
- Organizations
- Custom objects

USAGE OF BULK DELETE API

Bulk Delete is supported via the ROQL tabular queries and the syntax of a Bulk Delete API call in Connect REST is as follows:

```
https://your_site_interface/services/rest/connect/version/queryResults/?query=semicolon-separated  
ROQL DELETE queries
```

The syntax for a DELETE query is defined as follows:

```
DELETE FROM <primary Object> WHERE <where clause expression > ORDER BY <fields> LIMIT  
<number of rows>
```

Here are some example use cases for Bulk Delete API and corresponding ROQL Delete queries:

- Delete all obsolete incidents in the database that were created older than 1 Jan 2017 :
 - DELETE FROM Incidents WHERE CreatedTime < '2017-01-01T00:00:00Z' LIMIT 1000;
- Delete all opportunities in the database where ID < 100000
 - DELETE FROM Opportunities WHERE ID >= 1 AND ID < 1000; DELETE FROM Opportunities WHERE ID >= 1000 AND ID < 2000; DELETE FROM Opportunities WHERE ID >= 2000 AND ID < 3000; DELETE FROM Opportunities WHERE ID >= 3000 AND ID < 4000;...DELETE FROM Opportunities WHERE ID >= 19000 AND ID < 20000;
- For complex queries, perform a SELECT to fetch the IDs and pass the IDs as input in the DELETE statement.
 - Ex. DELETE from primary object where ID in (comma-separated list of IDs);

SUMMARY

Additional information on Bulk Delete and examples can be found in the API documentation:

- [Bulk Delete \(Connect REST\)](#)
- [Bulk Delete \(CWSS\)](#)
- [Bulk Delete \(Connect PHP\)](#)

Using custom objects for logging

The transactions table captures some of the field level changes on an object. However not all field changes are logged by default. Some customers use the combination of object event handler and custom objects in order to record the detailed level transactions (i.e. previous and current value) on existing and additional fields, as needed. In addition, custom objects are typically used to store the logs that are generated from a logging API. Please be thoughtful about the number of records that can be generated out of this custom logging.

It is advisable to have a purging process in place to delete the log entries (ps\$log table and/or any other custom objects logging table) on a regular basis. DLM can be used to delete the custom object entries based on a schedule.

Why are large database reductions not being seen?

One of the frequently asked questions is “Why hasn’t the database size reduced even after purging “X” million records?” It is because of how the database fragmentation works. When “X” number of rows are deleted from the tables in the database, the space left behind is empty. The empty space would be reclaimed as the new data is added. However, the entire empty space wouldn’t be reclaimed until database maintenance operations such as an upgrade or clone is performed on the site and the tables are optimized. Hence, the reduction in database size cannot be noticed immediately but will be seen over time.

High volume tables vs. course of action

Here is a summary of tools available to reduce the size of high volume tables:

High volume tables vs. course of action

HIGH VOLUME TABLE	RESOLUTION
incidents	Use DLM/Bulk Delete APIs to archive/purge large volume of incidents periodically.
threads	Deleting incidents using DLM/Bulk Delete APIs or archiving incidents will delete the corresponding thread records of incidents.
phrases	Apart from the config setting available for phrases table (PURGE_DELETE_INCIDENT_PHRASES), decrease the value of ph_size config verb if search is not frequently used. In addition, deleting/archiving the incidents will in turn delete the corresponding records from phrases table.
transactions	Use DLM to delete incident and contact transactions. Alternatively, deleting top-level objects will remove the corresponding transaction records from the table.
user_trans	Use the config verb PURGE_DELETE_USER_TRANS to

	define the purge policy for user_trans table.
co_trans	Similar to transactions table, deleting custom object records will remove the corresponding transaction records from co_trans table.
inc_performance	Use the config verb PURGE_DELETE_INC_PERFORMANCE to define the purge policy for inc_performance table. Alternatively, deleting/archiving the incidents will in turn delete the corresponding records from inc_performance table.
papi_meters/papi_method_meters	Starting 17D release and above, only the compliance data is retained in the transactional database for papi_meters and papi_method_meters tables. The transactional data is stored in another storage (with a pre-defined purge policy of 7 days) and is still available via reports for any debugging or troubleshooting scenarios.
clickstreams	Use the config verb PURGE_SESSION_TRACKING to define the purge policy for clickstreams table
contacts	Use DLM/Bulk Delete APIs to purge large volume of contacts periodically.
archived_incidents	Use the config verb PURGE_ARCHIVED_INCIDENTS to define the purge policy for archived_incidents table.
custom objects	Use DLM/Bulk Delete APIs to delete unwanted custom object records.
ps\$log	ps\$log can be treated like any other custom object. Make use of DLM/Bulk Delete APIs to delete unwanted records from ps\$log table.

Deleting top-level objects using any of the data management tools, will in turn delete all the related sub-objects from the database. For example, purging/archiving incidents will have an effect on the following tables: threads, phrases, transactions, inc_performance and other related sub-tables.

FILE ATTACHMENT SERVER (FAS) VOLUME MANAGEMENT

The File Attachment Server (FAS) is a remote file server used by the Oracle Service Cloud sites to store file attachments, archived incidents and file attachments associated with archived incidents (i.e. archive attachments). It is critical that the FAS volume is also monitored and kept in check.

Provided that the archived incidents are stored as XML files in FAS, make use of the configuration setting `PURGE_ARCHIVED_INCIDENTS` to actively purge the archived incident records from the database which in turn delete the corresponding XML files from FAS.

You can also limit the size of the file attachment at the time of upload via the `FATTACH_MAX_SIZE` configuration setting, thus controlling the size of files that can be uploaded on FAS.

Alternatively, delete the respective top-level or parent objects to which the file attachments are associated, to bring down the FAS volume.

BEST PRACTICES

Follow the best practices listed below to deliver a greater user experience for both agents and customers and to maintain an optimized Service Cloud database:

- Define and configure data volume management strategy based on business requirements and your organization's compliance policies. Based on the analysis, define and enforce lifecycle policies proactively than reactively on objects. Consider revisiting the data retention policies periodically to make sure they are in alignment with the requirements and business
- Monitor the database usage for your site regularly
- Consider purging data in `inc_performance`, `phrases`, `user_trans` and `session_summary` tables more aggressively using the available configuration verbs
- Pay more attention to incident data volume management
 - Archive and purge incidents more aggressively
 - Since archiving works only for incidents that are closed, close any incidents that have not been updated in a while
 - Purge from the archive which will in turn help clean up all the associated attachments from FAS
 - Make data retention schedules consistent across all interfaces
 - Make use of the Bulk Delete APIs to selectively delete large number of records that are obsolete
- Use Oracle Service Cloud primarily for transactional activity with customers
- Do not use Oracle Service Cloud as a data warehouse
- Consider moving any records that must be retained longer to an external storage (data warehousing), that allows for long-term retention of historical data and in turn improves the storage capacity of transactional database
- Consider recording the transactions for custom objects only when it is an absolute requirement.

CONCLUSION

This whitepaper outlined different ways to address the data volume management problems in Oracle Service Cloud.

If you need more help with any of the data volume management solutions, please contact your Oracle Account Team or Oracle Service Cloud Customer Care for recommendations on potential next steps.

There are many more enhancements coming up in the near future Service Cloud releases with respect to data volume management aspects. Please watch out for release notes and updates.

ADDITIONAL RESOURCES

Here are some of the KB articles that deals with some of the data volume management aspects outlined in this whitepaper.

- https://cx.rightnow.com/app/answers/detail/a_id/8492
- https://cx.rightnow.com/app/answers/detail/a_id/2362
- https://cx.rightnow.com/app/answers/detail/a_id/1234
- https://cx.rightnow.com/app/answers/detail/a_id/781
- https://cx.rightnow.com/app/answers/detail/a_id/10013

There are several other useful resources available to learn more about the features offered by Service Cloud, API guide and access to the Service Cloud Community as listed below:

- [Product Documentation and Tutorials](#)
- [API Documentation](#)
- [Service Cloud Community](#)

ORACLE CORPORATION

Worldwide Headquarters

500 Oracle Parkway, Redwood Shores, CA 94065 USA

Worldwide Inquiries

TELE + 1.650.506.7000 + 1.800.ORACLE1

FAX + 1.650.506.7200

oracle.com

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com. Outside North America, find your local office at oracle.com/contact.

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

Integrated Cloud Applications & Platform Services

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0219

White Paper Title

January 2017

Author: [OPTIONAL]

Contributing Authors: [OPTIONAL]



Oracle is committed to developing practices and products that help protect the environment



ORACLE®