Oracle Audit Vault

Best Practices

Nov 2007

Introduction

Oracle Audit Vault automates the audit data consolidation and analysis process, turning audit data into a key security resource to help address today's security and compliance challenges. Oracle Audit Vault is built on Oracle's industry leading database security and data warehousing products. This paper provides best practices for deploying Oracle Audit Vault in your enterprise. Information on deployment architectures and expected performance is included. In addition, this paper provides information on the auditing capabilities of the Oracle database and recommended best practices. Oracle Audit Vault supports consolidating audit data from Oracle9i Release 2 and higher databases. Oracle is currently working to support heterogeneous databases in a future release of Oracle Audit Vault.

Please note that this document will be updated on a regular basis to contain the latest information based on development and customer feedback. These best practices will be included in future releases of the Oracle Audit Vault documentation.

Installing Audit Vault

The architecture of Audit Vault consists of two major components that work in concert to store and secure the audit data. They are:

- *Audit Vault Server* – A stand-alone stacked application that contains a data warehouse built on a customized installation of Oracle Database 10g (10.2.0.3) with Oracle Database Vault providing security and OC4J components to support an Audit Vault Console and Enterprise Manager's Database Control.
- *Audit Vault Collection Agent* – The Collection Agent is responsible for managing collectors and maintaining the Audit Vault wallet.
    - *Collectors* - A collector is specific to an audit source and acts as the middleman between the source and the Audit Vault Server by pulling the audit trail data from the source and sending it to the Audit Vault Server over SQL*Net.
    - *Audit Vault Wallet* – The wallet is used to maintain the password for the collector to connect to the sources to pull audit data from the database.

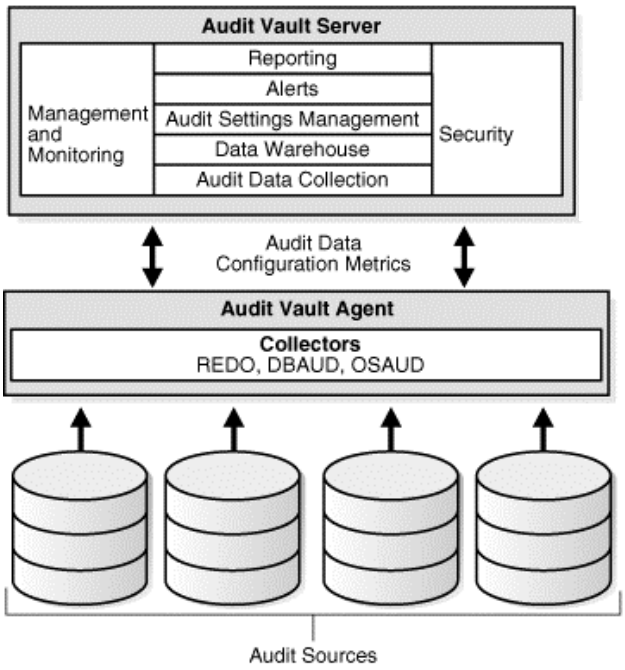


Figure 1 Audit Vault Architecture

Deployment Plan

While Audit Vault provides consolidation and secure storage of audit data, planning the installation of the Audit Vault components will ensure a faster installation and overall success of implementing a compliant solution. The following sections discuss the pre-installation considerations for the Audit Vault Server and Audit Vault Collection Agents.

Audit Vault Server

The Audit Vault Server should be installed on its own host or a host that contains other repository databases such as Enterprise Manager Grid Control or the Oracle Recovery Manager (RMAN) repository database. By installing the Audit Vault Server separate from the source database servers provides the following benefits:

- Higher Availability – When the Audit Vault Server is on a separate server from the source databases then the availability will not be dependent on the source host's up/down status and therefore the audit data continues to be collected from all sources that are running.
- Secured Audit Trail – By extracting the audit trail records off of the source database as fast as possible, there is very little opportunity for privileged database and operating system users to modify any audit records.

When it comes to determining what type of resources are required to install and maintain the Audit Vault Server, it depends on the how fast you need the audit records to be inserted into Audit Vault and how long you must maintain audit data.

In internal testing on a 2x6GB 3GHz Intel Xeons, Redhat 3.

2 Linux host, the Audit Vault Server inserted up to 17,000 audit records / second. To store 500,000 audit trail records in the Audit Vault repository database requires approximately 300mg of disk space. An additional 2G of disk space is needed for the ORACLE_HOME files.

For scalability and availability, the Audit Vault Server may optionally implement Real Applications Cluster (RAC) and Data Guard for disaster recovery.

Check the specific Audit Vault Server Installation Guide documentation of the platform that you will be installing for a list of the requirements of that operating system.

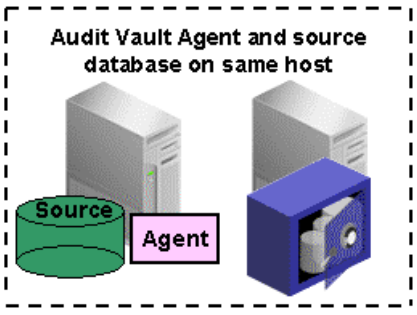
Audit Vault Collection Agent

The Oracle database can write audit trail data into the database (SYS.AUD$/SYS.FGA_LOG$) and/or operating system files. The online log (redo log) of the Oracle database also contains information of before/after value changes of data as well. Audit Vault deploys a process called a Collector which is specific to the Oracle database audit trail to extract the audit data and send it to the Audit Vault Server. The three types of collectors are called DBAUD for database auditing, OSAUD for operating system files written by the Oracle database, and REDO to extract audit data from the redo stream.
The Audit Vault Collection Agent provides support for audit data collection. The agent loads the collectors, provides them with a connection to the Audit Vault Server to send audit data and run-time metrics on the collectors. Audit Vault communicates with the audit data source through its agent

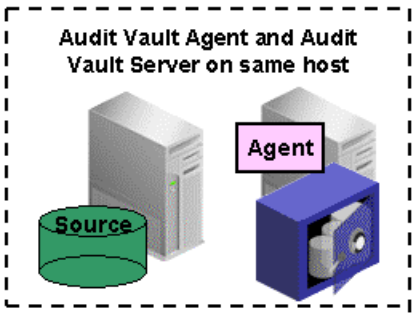
The Audit Vault Collection Agent may be installed either on the same host as the database that is going to be audited, on the audit vault server hosts, or on a host separate from the audit vault server or the host where the database resides that will be audited.

Let's look at each of these scenarios to determine the best location within your environment for the Audit Vault Collection Agent.

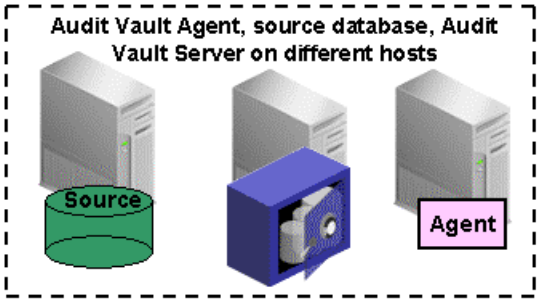
- **Same host of audited databases (Recommended)** – If the database audit trail destination is the operating system, the Audit Vault Collection Agent must be installed on the same hosts as those operating system files.



- **Audit Vault Server host** – If the database audit trail destination is the database tables (SYS.AUD$/SYS.FGA_LOG$) then the Audit Vault Collection Agent may be installed on the Audit Vault Server host. This would mean that all software components used by Audit Vault would be consolidated on a single host.



- **Separate from audit host and Audit Vault Server** – If the database audit trail destination is the database tables, (SYS.AUD$/SYS.FGA_LOG$) then the Audit Vault Collection Agent may be installed on a different host from the audited database or Audit Vault Server.



Recommended Agent Configuration

Oracle recommends that the Audit Vault Collection Agent be installed on the same server as the databases being audited. In the case of RAC the agent should be installed on each instance. This configuration will allow the agent to service audit data from either the database tables (SYS.AUD$/SYS.FGA_LOG$) or the operating system files.

Which Collector(s) Should I Deploy?

Audit Vault collectors transport audit data from the source to the Audit Vault Server. The collectors are controlled by the Audit Vault Collection Agents described in the previous section. Oracle Audit Vault Collection Agent may deploy three different Audit Vault collectors depending on where the audit data is written - database tables or operating system. Note that Oracle stores some valuable audit related information in the REDO logs. As a result, Oracle Audit Vault provides a REDO Collector to retrieve the information. Table 1 below lists the characteristics of the audit trail locations to help

you determine where to write the audit trail and which collector(s) should be deployed to move the audit data into Audit Vault.

Audit Operation	OS Log	DB Audit Table	Redo Log
SELECT	✓	✓	
DML	✓	✓	✓
DDL	✓	✓	✓
Before and After Values			✓
Success and Failure	✓	✓	
SQL Text	✓ (for SYS)	✓	
SYS Auditing	✓		✓
Other considerations	Separation of Duties	FGA data	Supplemental logging for all values

Table 1 Audit Trail Characteristics

The three collector types are called DBAUD, OSAUD, and REDO. Each collector type retrieves audit records from different locations in the source Oracle database as shown below in Table 2.

Collector Name	Audit Data Sources of Oracle Database	Oracle Database Settings to Initiate Auditing	Advantages
DBAUD	Database audit trail, where standard audit events are written to the database dictionary table SYS.AUD$. Fine-grained audit trail, where audit events are written to the database dictionary table SYS.FGA_LOG$	Set initialization parameter audit_trail=db, db_extended.	With the DB_EXTENDED value, the SQL text is collected as part of the audit trail. DB_EXTENDED does not audit activity by SYS users, so you should also deploy the OSAUD collector in conjunction.
OSAUD	Operating system files (OS files), where mandatory	Set initialization parameter AUDIT_TRAIL parameter to OS	• Audit records stored in operating system

Collector Name	Audit Data Sources of Oracle Database	Oracle Database Settings to Initiate Auditing	Advantages
	audit records are written and optionally, where Database audit trail (standard audit events) and fine-grained audit trail audit events are written to OS audit logs. Operating system-specific audit trails (system audit trail), where database audit trail records are written to Windows Event Log on Microsoft Windows systems or to a syslog on Linux systems	and AUDIT_FILE_DEST parameter to desired file in directory specification. Set initialization parameters AUDIT_SYS_OPERATIONS to TRUE and AUDIT_FILE_DEST to the desired file in the directory.	files can be more secure than database-stored audit records because access can require file permissions that DBAs do not have. • Greater availability is another advantage to operating system storage for audit records, in that they remain available even if the database is temporarily inaccessible.
REDO	Redo log	Redo logs are part of the Oracle Database infrastructure and do not require any source database settings. The Audit Vault Policy, capture rule, determines the meta data pulled from the redo log.	• Used to track before and after changes to sensitive data columns, such as salary.

Table 2 Audit Vault Collector Types

Depending on the type of audit information generated and required to maintain, you may deploy one or all three of the collectors for each source database.

Recommended Collector and Database Audit Configuration

Oracle recommends using the operating system as your as your primary audit trail location and deploying the OSAUD collector as the operating system has the least amount of performance overhead on the database. Please refer to the Oracle Database Auditing section within this document for information on configuring the database to write audit information to the operating system.

Near Real-Time Alerts

Security alerts can be used for proactive notification of compliance, privacy, and insider threat issues across the enterprise. Oracle Audit Vault provides IT security personnel with the ability to detect and alert on suspicious activity, attempts to gain unauthorized access, and abuse of system privileges.

Oracle Audit Vault can generate alerts on specific system or user defined events, acting as an early warning system against insider threats and helping detect changes to baseline configurations or activity that could potentially violate compliance. Oracle Audit Vault continuously monitors the audit data collected, evaluating the activities against defined alert conditions.

Alerts are generated when data in a single audit record matches a custom defined alert rule condition. For example, a rule condition may be defined to raise alerts whenever a privileged user attempts to grant someone access to sensitive data.

In Oracle's in-house testing of the Audit Vault Server, it was possible to achieve a throughput of 17,000 insertions of audit trail records per second using a 2x6GB 3GHz Intel Xeons, Redhat 3.
2 Linux x86 system. To achieve near real-time alerting capability, the host should be sized to meet your business requirements.

Near Real-Time Reporting

After audit data is transferred from the source to the Audit Vault, an Oracle DBMS_SCHEDULER job runs an ETL (extract, transformation, load) process to normalize the raw audit data into the data warehouse. In Oracle's in-house testing, the ETL job was able to process 500,000 records in a little over 50 seconds on a 2x6GB 3GHz Intel Xeons, Redhat 3.
2 Linux x86 system. Out of the box, the default DBMS_SCHEDULER job runs every 24 hours.

Audit Vault provides statistics of the ETL process to update the warehouse as shown below in Figure 3. By utilizing this information, you can estimate how often the job may be run to update the data warehouse infrastructure. The data warehouse infrastructure is documented in the Oracle Audit Vault Auditor's Guide.

Recommendations on ETL process

The ETL process may be run more often to provide near real-time reporting. Oracle recommends that the previous ETL job be completed before initiating the next ETL job.

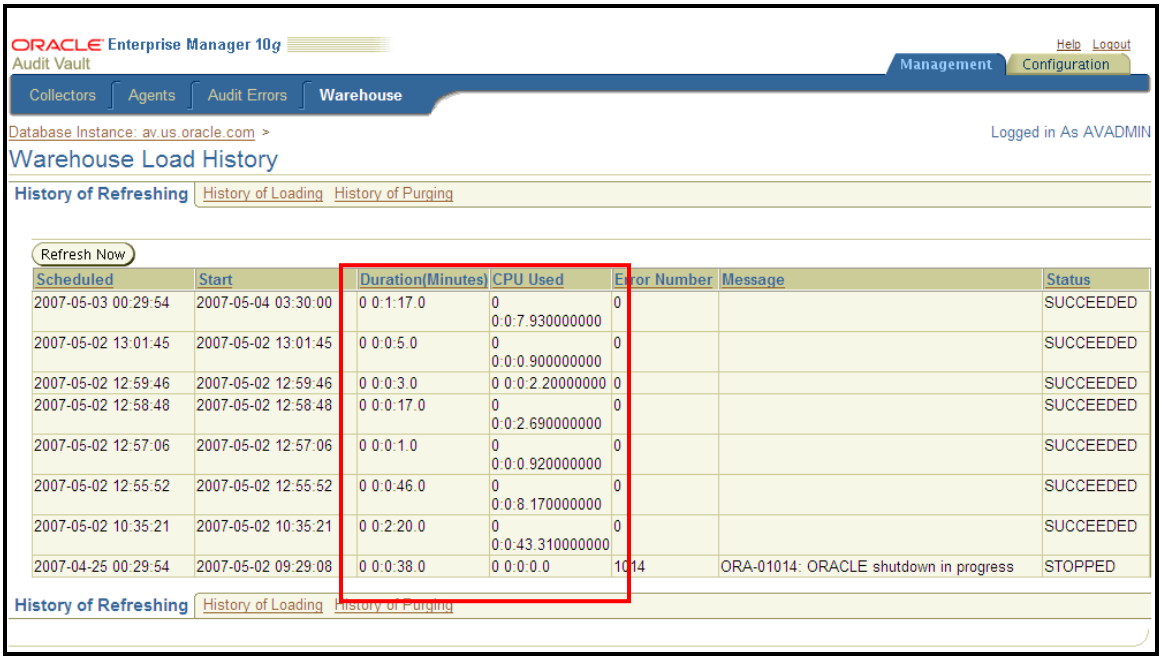


Figure 2 Audit Vault Warehouse Load Results

The Oracle Audit Vault has been developed on a flexible data warehouse infrastructure that provides the ability to consolidate audit data so that it can be easily secured, managed, accessed, and analyzed. In addition to the out-of-the-box reports provided by Oracle Audit Vault, Audit Vault provides an open audit warehouse schema that can be accessed from Oracle BI Publisher, Oracle Application Express, or any 3rd party reporting tools for customized security and compliance reporting.

Oracle Database Auditing

Oracle has provided robust auditing capabilities since the release of Oracle7 in the early 1990's. Oracle database auditing can be highly customized to address specific compliance and privacy requirements.

Audit records include information about the operation that was audited, the user performing the operation, and the date and time of the operation. Audit records can be stored in the database audit trail or in files on the operating system. There are two types of general auditing: standard and fine-grained. Standard auditing includes operations on privileges, schemas, objects, and statements. Fine-grained auditing is policy based and operates and is enforced on select operations in Oracle9i. Fine-grained auditing was enhanced in Oracle Database 10g to enforce policy based auditing on insert, update and delete operations.

Audit Trail Contents and Locations

Audit trail records can contain different types of information, depending on the events audited and the auditing options set.

Some of that information includes:

- Operating system login user name (CLIENT USER)
- Database user name (DATABASE USER)
- Session identifier
- Terminal identifier
- Name of the schema object accessed
- Operation performed or attempted (ACTION)
- Date and time stamp in UTC (Coordinated Universal Time) format
- System privileges used (PRIVILEGE)
- Proxy Session audit ID
- Global User unique ID
- Instance number
- Process number
- Transaction ID
- SCN (system change number) for the SQL statement
- SQL text that triggered the auditing (SQLTEXT)
- Bind values used for the SQL statement, if any (SQLBIND)

Audit Vault extracts auditdata from either the database tables or the operating system files. To enable database auditing, the initialization parameter, AUDIT_TRAIL, should be set to one of these values:

Parameter Value	**Meaning**
DB	Enables database auditing and directs all audit records to the database audit trail (SYS.AUD$), except for records that are always written to the operating system audit trail
DB_EXTENDED	Does all actions of AUDIT_TRAIL=DB and also populates the SQL bind and SQL text columns of the SYS.AUD$ table,
OS (recommended)	Enables database auditing and directs all audit records to an operating system file

Recommended Database Audit Configuration

Oracle recommends that the audit trail be written to the operating system files as this configuration imposes the least amount of overhead on the source database system.

In addition, the following database parameters should be set:

- Init.ora parameter: AUDIT_FILE_DEST -- Dynamic parameter specifying the location of the operating system audit trail. The default location on Unix/Linux is $OH/admin/$ORACLE_SID/adump. The default on Windows is the event log. For optimal performance, it should refer to a directory on a disk that is locally attached to the host running the Oracle instance.
- Init.ora parameter: AUDIT_SYS_OPERATIONS -- Enables the auditing of operations issued by user SYS, and users connecting with SYSDBA or SYSOPER privileges. The audit trail data is written to the operating system audit trail. This parameter should be set to true.

Audit Settings – Secure Configuration

Oracle database auditing is highly granular, flexible and extensible. In most enterprise environments, auditing of basic activities such as failed and successful logins, privileged user activity, database schema changes, and user policy changes will be required by IT auditors.

When you issue an audit command, an additional parameter 'by access' or 'by session' can be specified. *By access* tells Oracle to create an audit record every time any of these operations occur when in contrast, *by session* only creates an audit trail the first time this operation occurs in the current session. If you need to know each time an operation is executed then *by access* should be used.

Recommended Database Audit Settings

Oracle recommends the following audit settings for your source databases to collect information on the operations executed. A SQL script can be found in Appendix B that may be copied and run in your databases. When Audit Vault is installed, this script is also included in the demo directory of the Audit Vault Server $ORACLE_HOME/demo/secconf.sql.

Audit Command	What do you audit?

`Audit alter any table by access;`
`Audit create any table by access;`
`Audit drop any table by access;`
`Audit Create any procedure by access;`
`Audit Drop any procedure by`
	Database schema or structure changes

access; Audit Alter any procedure by access; Audit create external job by access; Audit create any job by access; Audit create any library by access; Audit alter database by access; Audit alter system by access;	
Audit audit system by access; Audit create public database link by access; Audit exempt access policy by access; Audit alter user by access; Audit create user by access; Audit role by access; Audit create session by access; Audit drop user by access; Audit Grant any privilege by access; Audit grant any object privilege by access; Audit grant any role by access; Audit alter profile by access; Audit drop profile by access;	Database access and privileges

Table 3 Recommended Audit Settings

TIP: Do not audit the SYS.AUD$ or SYS.FGA_LOG$ tables. This will cause a recursive condition.

Oracle also has the ability to create specific audit policies based on a condition called Fine-Grained Auditing. By utilizing fine-grained auditing, you can monitor data access based on content or condition. Conditions can include limiting the audit to specific types of DML statements used in connection with the columns that you specify. Optionlally a named routine can be called when an audit event occurs to handle errors and anomalies.

An example of a fine-graned audit policy to create an audit trail record if a select on the SH.SALES table is executed by anyone other than the user APPS is shown in Figure 2.

Based on your business requirements, fine-grained auditing can be tailored to meet the auditing needs. For more information on database auditing, please see the Oracle Security Guide documentation.

Figure 3 Audit Vault Fine Grained Audit Policy Example

Database Auditing Performance

On the source database, resources on required by the audit process and the Audit Vault Collection Agent.

Auditing and the Audit Vault Collectors

Using the recommended audit settings listed above in Table 3, Oracle performed in house testing on a 4x32GB 3GHz Intel Xeons Redhat 3.0, running
Oracle Database 10g Release 2 (10.2.0.3). The table below demonstrates that database auditing and Audit Vault Collection Agent uses up to 6% additional CPU overhead based on the number of audit trail records created per second.

	10 audit/sec		100 audit/sec	
	Create;	Collect	Create;	Collect
OS Log	0.03%;	0.7% (c)	0.07%,	2.7% (c)
DB Audit	0.29%;	0.5% (c)	2.4%;	1.7% (c)
Redo	0;	3.7% (c)	0;	5.9% (c)

Table 4 Auditing and Collection Agent CPU Overhead

Table 4 shows the performance overhead of turning on auditing and running a TPC-C like workload using the recommended audit settings specified in Table 3. The 'Collect' column shows the performance overhead of the specific Audit Vault Collector while the 'Create' number shows the performance overhead for database auditing when 10 or 100 audit records are generated per second.

Writing audit trail records to the operating system has the lowest overhead.

Managing Audit Data on the Source

Audit records include information about the operation that was audited, the user performing the operation, and the date and time of the operation. As noted earlier, audit records can be stored in either the database or on the operating system**.**

The tables named SYS.AUD$ and SYS.FGA_LOG$ are used when the audit data is written to the database. These tables are located in the databse SYS schema.

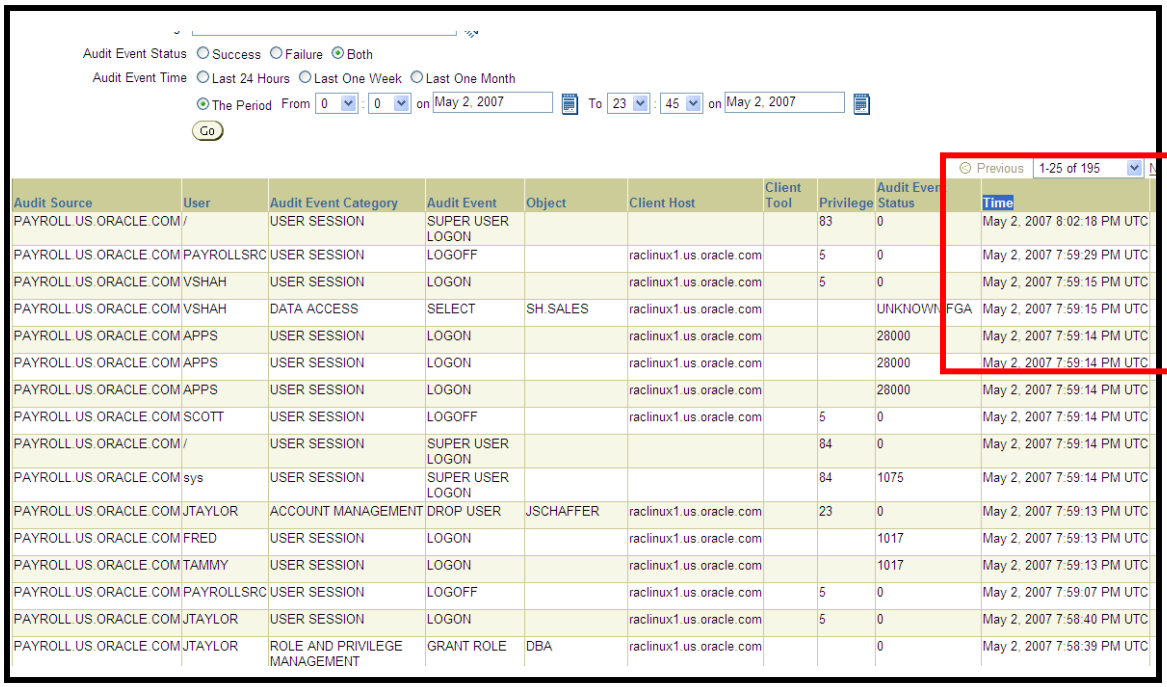
The Oracle Database also allows audit trail records to be directed to the operating system. The target directory varies by platform, but on the UNIX platform, it is usually $ORACLE_HOME/rdbms/audit. On Windows, the information is accessed through Event Viewer.

Oracle Audit Vault provides the mechanisms to collect audit data generated by Oracle9*i* Database Release 2, Oracle Database 10g Release 1, and Oracle Database 10*g* Release 2. The database audit data can be collected from the both the database and operating system audit destinations. Transactional before/after values can be captured from the database REDO transaction logs using the REDO collector for Oracle9i Release 2 and Oracle Database 10g Release 2 databases.

Removing Audit Data from the Database

Over time, the database and operating system can potentially reach a maximum capacity for storing new audit records After auditing is enabled for some time, the security administrator will want to delete records from the database audit trail both to free audit trail space and to facilitate audit trail management. However, it's critical not to delete data that has not yet been transferred to Oracle Audit Vault.

Before deleting audit data from the database, determine the last record inserted into Audit Vault Server. This can be done by using Audit Vaults's Activity Overview Report. Open the Activity Overview to view the date of the summary data. Remember, the Audit Vault report data is displayed based on the last completed ETL warehouse job. For more information on the warehouse job, please look at the Oracle Audit Vault Administration Guide documentation.



Audit Event Status ○ Success ○ Failure ◉ Both

Audit Event Time ○ Last 24 Hours ○ Last One Week ○ Last One Month

◉ The Period From 0 : 0 on May 2, 2007 To 23 : 45 on May 2, 2007

Go

Previous 1-25 of 195

Audit Source	User	Audit Event Category	Audit Event	Object	Client Host	Client Tool	Privilege	Audit Event Status	Time
PAYROLL.US.ORACLE.COM	/	USER SESSION	SUPER USER LOGON				83	0	May 2, 2007 8:02:18 PM UTC
PAYROLL.US.ORACLE.COM	PAYROLLSRC	USER SESSION	LOGOFF		raclinux1.us.oracle.com		5	0	May 2, 2007 7:59:29 PM UTC
PAYROLL.US.ORACLE.COM	VSHAH	USER SESSION	LOGON		raclinux1.us.oracle.com		5	0	May 2, 2007 7:59:15 PM UTC
PAYROLL.US.ORACLE.COM	VSHAH	DATA ACCESS	SELECT	SH.SALES	raclinux1.us.oracle.com			UNKNOWN	FGA May 2, 2007 7:59:15 PM UTC
PAYROLL.US.ORACLE.COM	APPS	USER SESSION	LOGON		raclinux1.us.oracle.com			28000	May 2, 2007 7:59:14 PM UTC
PAYROLL.US.ORACLE.COM	APPS	USER SESSION	LOGON		raclinux1.us.oracle.com			28000	May 2, 2007 7:59:14 PM UTC
PAYROLL.US.ORACLE.COM	APPS	USER SESSION	LOGON		raclinux1.us.oracle.com			28000	May 2, 2007 7:59:14 PM UTC
PAYROLL.US.ORACLE.COM	SCOTT	USER SESSION	LOGOFF		raclinux1.us.oracle.com		5	0	May 2, 2007 7:59:14 PM UTC
PAYROLL.US.ORACLE.COM	/	USER SESSION	SUPER USER LOGON				84	0	May 2, 2007 7:59:14 PM UTC
PAYROLL.US.ORACLE.COM	sys	USER SESSION	SUPER USER LOGON				84	1075	May 2, 2007 7:59:14 PM UTC
PAYROLL.US.ORACLE.COM	JTAYLOR	ACCOUNT MANAGEMENT	DROP USER	JSCHAFFER	raclinux1.us.oracle.com		23	0	May 2, 2007 7:59:13 PM UTC
PAYROLL.US.ORACLE.COM	FRED	USER SESSION	LOGON		raclinux1.us.oracle.com			1017	May 2, 2007 7:59:13 PM UTC
PAYROLL.US.ORACLE.COM	TAMMY	USER SESSION	LOGON		raclinux1.us.oracle.com			1017	May 2, 2007 7:59:13 PM UTC
PAYROLL.US.ORACLE.COM	PAYROLLSRC	USER SESSION	LOGOFF		raclinux1.us.oracle.com		5	0	May 2, 2007 7:59:07 PM UTC
PAYROLL.US.ORACLE.COM	JTAYLOR	USER SESSION	LOGON		raclinux1.us.oracle.com		5	0	May 2, 2007 7:58:40 PM UTC
PAYROLL.US.ORACLE.COM	JTAYLOR	ROLE AND PRIVILEGE MANAGEMENT	GRANT ROLE	DBA	raclinux1.us.oracle.com			0	May 2, 2007 7:58:39 PM UTC

Figure 4 Audit Vault Activity Overview Report

The activity overview report returns data in descending order of time. So the first record displayed is the last record to be inserted into the data warehouse.

Once you have established that data is being inserted into the Audit Vault Server in a timely manner, you can use the scripts located in Appendix A to delete records from SYS.AUD$ and SYS.FGA_LOG$ by running a database job.

Recommended Database Audit Cleanup Periods

Oracle recommends that you should delete records 24 hours and older. In the example above, you would delete records that are older than May 3, 2007 8:02 PM UTC. All data is stored in Audit Vault in UTC time format to maintain the order in which transactions are executed no matter what time zone the database sources are located in.

Removing Audit Data from the Operating System

Similar to the database audit trail, Oracle stores audit data on the operating system by creating or appending to operating system files based on the Oracle database session id. Since disk space is not infinite, the operating system audit trail files will need to be deleted after the records are inserted in to Audit Vault.

The operating system audit trail files are written by default on most UNIX system in $ORACLE_HOME/admin/$ORACLE_SID/adump. The files have an extension of ".aud" Optionally, the destination can be explicitly defined by the Oracle database parameter AUDIT_FILE_DEST.

Before deleting audit trail records, determine the last record inserted into Audit Vault Server. This can be done by using the Activity Overview Report. Open the Activity Overview to view the date of the summary data (See Figure 4 above). Remember, the Audit Vault report data is displayed based on the last completed ETL warehouse job. For more information on the warehouse job, please look at the Oracle Audit Vault Administration Guide. Appendix A contains scripts that can be run as a cron job or database job to delete operating system audit files that are no longer needed on UNIX systems.

On the Windows operating system, the audit trail record is written to the window event log. Use the Windows Event Viewer functionality to control the size of the event log file or overwrite records that are older the X number of days.

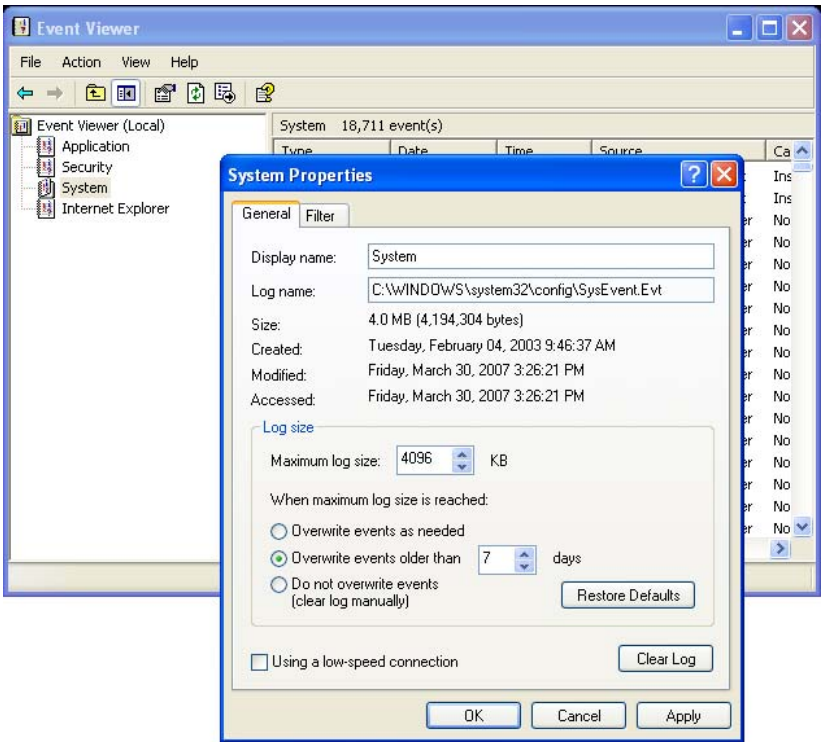


Figure 5 Windows Event Viewer

Oracle recommends that you should use the option to overwrite records based on age.

Oracle Audit Vault Maintenance

Periodic maintenance of Oracle Audit Vault is important for maintaining optimal performance. Oracle Audit Vault generates numerous logs and trace files during normal day-to-day operations. The following sections provide important information regarding the contents of the log files, their purpose and how and when the files can be removed.

Audit Vault Server Log Files

Much like the Oracle Database, the Oracle Audit Vault server generates log files that provide current status and diagnostic information. The log files should be monitored and periodically removed to control the amount of disk space used by the log files. These log files may be found in *<Audit_Vault_Server_Home>*/av/log.

Server Log File Name	Description	Maintenance
avorcldb.log	This log file tracks the commands issued by the avorcldb facility. Avorcldb facility is used during the initial configuration of audited sources and Audit Vault agents and collectors.	It is safe to delete this file at any time.
avca.log	This log file tracks the creation of collectors and the starting and stopping of Audit Vault agents and collectors.	This file may only be deleted after the Audit Vault Server is shutdown.
av_client-%g.log.n	This log file contains information about collection metrics from the Audit Vault Collection Agent. The %g is a generation number that starts from 0 (zero) and increases once the file size reaches the 10 MB limit.	The files, which contain an extension of .log.n, for example av_client-0.log.1, may be deleted at any time.

Enterprise Manager stores its logs in the directory *<Audit Vault_Server_Home>/<Host_Name>_<SID>/sysman/log*. The file emdb.nohup in this directory contains a log of activity for the Audit Vault web application, including GUI conversations, requests from the *avctl* utility and communication with the various Audit Vault collection agents. This can be used to debug communication issues between the server and the agents.

Audit Vault Collection Agent Log Files

The Audit Vault Collection Agent creates several log files and also must be maintained to control the amount of disk space used by the log files. These log files may be found in *<Audit_Vault_Collection_Agent_Home>*/av/log.

Agent Log File Name	Description	Maintenance
agent.err	Contains a log of all errors encountered in agent initialization and operation.	It is safe to delete this file at any time.
agent.out	Contains a log of all primary agent-related operations and activity.	This file may only be deleted after the Audit Vault Collection Agent is shutdown.
avca.log	Contains a log of all AVCA commands that have been run and the results of running each command.	It is safe to delete this file at any time.
avorcldb.log	Contains a log of all AVORCLDB commands that have been run and the results of running each command.	It is safe to delete this file at any time.
<CName><SName><SId>.log CNmae = Collector_name SName = Source_name SID = Source_ID	Contains a log of collection operations for the DBAUD and OSAUD collectors.	This file may only be deleted after the Audit Vault Collection Agent is shutdown.
av_client-%g.log.n	Contains a log of the agent operations and any errors returned from those operations. The %g is a generation number that starts from 0 (zero) and increases once the file size reaches the 10 MB limit. A concurrent existence of this file is indicated by a .n suffix appended to the file type name, such as av_client-%g.log.n, where n is an integer issued in sequence, for example av_client-0.log.1.	The files which contain an extension of .log.n may be deleted at any time.
sqlnet.log	Contains a log of SQL*Net information.	

The directory *<Audit_Vault_Collection_Agent_Home>/oc4j/j2ee/home/log* contains the logs generated by the Collection Agent OC4J. In this directory, the file AVAgent-access.log contains a log of requests the agent receives from the Audit Vault Server. This can be used to debug communication issues between the server and the agent.

Oracle Audit Vault Disaster Recovery

By default, the Oracle Audit Vault data warehouse is in archive log mode. This protects the audit data from media failure and ensures a more complete recovery. The archive logs are placed in the flash recovery area.

Oracle Recovery Manager (RMAN) and a flash recovery area minimize the need to manually manage disk space for your backup-related files and balances the use of space among the different types of files. The basic Oracle Audit Vault installation places the flash recovery area on the same disk as the Audit Vault Oracle Home and sets the default size to 2G. The advanced installation method allows you to define the location and size of the flash recovery area and RMAN backup job.

Recommended Recovery Configuration

Oracle recommends that you review the flash recovery area settings and modify them to meet your data protection needs. For more information on RMAN, flash recovery area, and archive logs, please see the Oracle Database Backup and Recovery documentation. The Audit Vault Oracle Homes should be backed up using your current procedures for other Oracle Homes.

Appendix A. Audit Trail Maintenance Scripts

```
Rem
Rem $Header: os_aud_cleanup_setup.sql 05-apr-2007.02:42:14 srirasub Exp
$
Rem
Rem os_aud_cleanup_setup.sql
Rem
Rem Copyright (c) 2007, Oracle. All rights reserved.
Rem
Rem    NAME
Rem      os_aud_cleanup_setup.sql - <one-line expansion of the name>
Rem
Rem    DESCRIPTION
Rem      <short description of component this file declares/defines>
Rem
Rem    NOTES
Rem      <other useful comments, qualifications, etc.>
Rem
Rem    MODIFIED   (MM/DD/YY)
Rem    srirasub    04/04/07 - Created
Rem

-- The following arguments are required to run this procedure
--
--  1. Path of directory where temporary files can be written (eg.
/tmp)
--  2. Threshold (in no. of days) for deleting old audit files (eg. 7)
--  3. $ORACLE_HOME
--
--
SET ECHO ON
SET FEEDBACK 1
SET NUMWIDTH 10
SET LINESIZE 80
SET TRIMSPOOL ON
SET TAB OFF
SET PAGESIZE 100

create or replace procedure source_os_audit_cleanup as
  output_file  utl_file.file_type;
  cursor c1 is select unique(audsid) from sys.V$SESSION;
  sessid number;
  aud_dest varchar2(1000);
  ver varchar2(100);
begin
  execute immediate 'create or replace directory os_aud_cleanup_dir as
''&1''';

  output_file := utl_file.fopen
('OS_AUD_CLEANUP_DIR','session_list.txt', 'W');
  open c1;
  loop
    fetch c1 into sessid;
    exit when c1%notfound;
```

```
    utl_file.put_line (output_file, sessid);
  end loop;
  utl_file.fclose(output_file);

  select value into aud_dest from v$parameter where name =
'audit_file_dest';

  select version into ver from v$instance;

  if ver like '10%' or ver like '11%'
  then
    execute immediate
      'BEGIN dbms_scheduler.run_job(''OS_CLEANUP_PERL'',TRUE); END;';
  else if ver like '9%'
   then
     output_file := utl_file.fopen
('OS_AUD_CLEANUP_DIR','audit_dest.txt', 'W');
     utl_file.put_line (output_file, aud_dest);
     utl_file.fclose(output_file);
   end if;
  end if;
end;
/

Declare
  ver varchar2(100);
  argv3 varchar2(1000);
  argv2 varchar2(1000);
  aud_dest varchar2(1000);
Begin
  select version into ver from v$instance;
  select value into aud_dest from v$parameter where name =
'audit_file_dest';

  argv2 := '&1' || '/' || 'session_list.txt';
  argv3 := '&3' || '/demo/os_aud_cleanup.pl';

  if ver like '10%' or ver like '11%'
  then

  execute immediate
    'BEGIN DBMS_SCHEDULER.CREATE_JOB (JOB_NAME => ''OS_CLEANUP_PERL'',
                JOB_TYPE => ''executable'',
                JOB_ACTION => ''' || argv3 || ''',
                NUMBER_OF_ARGUMENTS => 3,
                ENABLED => FALSE); END;';

  execute immediate
    'BEGIN DBMS_SCHEDULER.SET_JOB_ARGUMENT_VALUE (
       job_name              => ''OS_CLEANUP_PERL'',
       argument_position     => 1,
       argument_value        => ''' || aud_dest || '''); END;';

  execute immediate
     'BEGIN DBMS_SCHEDULER.SET_JOB_ARGUMENT_VALUE (
       job_name              => ''OS_CLEANUP_PERL'',
       argument_position     => 2,
```

```
       argument_value         =>''' || argv2 || '''); END;';

  execute immediate
   'BEGIN DBMS_SCHEDULER.SET_JOB_ARGUMENT_VALUE (
       job_name               => ''OS_CLEANUP_PERL'',
       argument_position      => 3,
      argument_value         => ''&2''); END;';

  execute immediate
   'BEGIN DBMS_SCHEDULER.CREATE_JOB (JOB_NAME => ''AUDIT_OS_CLEANUP'',
                  JOB_TYPE => ''STORED_PROCEDURE'',
                  JOB_ACTION => ''sys.source_os_audit_cleanup'',
                  REPEAT_INTERVAL => ''FREQ=DAILY;INTERVAL=1'',
                  ENABLED => TRUE,
                  COMMENTS => ''Cleaup Job Run Daily''); END;';
  end if;
End;
/

declare
  ver varchar2(100);
begin
  select version into ver from v$instance;

  if ver like '9%'
  then
    sys.source_os_audit_cleanup;
  end if;
end;
/


exit;



#!/usr/local/bin/perl
#
# $Header: os_aud_cleanup.pl 05-apr-2007.01:47:12 srirasub Exp $
#
# os_aud_cleanup.pl
#
# Copyright (c) 2007, Oracle. All rights reserved.
#
#    NAME
#      os_aud_cleanup.pl - OS AUDit trail CLEANUP
#
#    DESCRIPTION
#      Perl Script to clean audit trails.
#
#    NOTES
#      <other useful comments, qualifications, etc.>
#
#    MODIFIED   (MM/DD/YY)
#    srirasub    04/04/07 - Creation
#
```

```
$aud_dir = $ARGV[0];

%mon2num = qw(
  jan 1  feb 2  mar 3  apr 4  may 5  jun 6
  jul 7  aug 8  sep 9  oct 10 nov 11 dec 12
);

#list of all files in audit dir
if(!opendir(DIR, $aud_dir))
{
  $oh = $ENV{'ORACLE_HOME'};

  $aud_dir =~ s/\?/$oh/g;

  if(!opendir(DIR, $aud_dir))
  {
    exit -1;
  }
}

@files = grep(/ora_.*aud$/,readdir(DIR));
closedir(DIR);

#get timestamp to compare
$tstamp = time();

$tstamp1 = localtime;

#get list of active stession from db
$session_list = $ARGV[1];
open(INFO, $session_list);
@sessids = <INFO>;
close(INFO);

#days parameter
$day_upper_limit = $ARGV[2];

#go thru all the files in audit destination directory
foreach $file (@files)
{
  $flag = 1;
  $file_name = $aud_dir. '/' . $file;
  open(INFO, $file_name);
  @lines = <INFO>;
  close(INFO);

  #check each line for matchin session
  foreach $line (@lines)
  {
    foreach $sess (@sessids)
    {
      $sessionid = $sess;
      chop($sessionid);
      $reg = 'SESSIONID: "' . $sessionid . '"';
      if($line =~ $reg)
      {
```

```
        #this file cant be deleted as it has a session that
        #is active
        $flag = 0;
      }
    }

    $prev = $line;
  }

  #since this file doesnt have any active session, it can be deleted
  if($flag == 1)
  {
    $flag2 = 1;
    foreach $line (@lines)
    {
      $reg = 'SESSIONID: "';

      if($line =~ $reg)
      {
        chop($prev);
        $days_diff = day_diff($prev, $tstamp1 );

        if($days_diff < $day_upper_limit)
        {
          #this file cant be deleted as it has a session that
          #doesnt satisfy the min-days criteria
          $flag2 = 0;
        }
      }
      $prev = $line;
    }

    if($flag2 == 1)
    {
      #delete the file
      unlink("$file_name");
    }
  }
}

#subroutine to return difference between two dates.
sub day_diff
{
  $ts1 = $_[0];
  $ts2 = $_[1];

  $ts1 =~ s/\s\s*/ /g;
  $ts2 =~ s/\s\s*/ /g;

  ($a1,$a2,$a3,$a4,$a5,$a6,$a7) = split(/[ :]/, $ts1);
  ($b1,$b2,$b3,$b4,$b5,$b6,$b7) = split(/[ :]/, $ts2);

  $diff_mon = &month_difference($a2, $b2);
  $days = (($b7-$a7)*365) + ($diff_mon)*30 + ($b3-$a3);
  $days;
}
```

```
#subroutine to return difference between two month
sub month_difference
{
  $mon1 = $mon2num{ lc substr($_[0], 0, 3) };
  $mon2 = $mon2num{ lc substr($_[1], 0, 3) };

  $diff = $mon2-$mon1;

  $diff;
}


-- These scripts can be run directly on 9i, 10gR1 and 10gR2
-- scripts should be run as sys
-- the jobs run daily

-- For DBMS_JOB,
-- ALTER SYSTEM SET job_queue_processes=1;
-- this may be required to run the jobs automatically
-- the value is set to 0 in most systems
-- any number in the range [1,1000] is valid.


create or replace procedure source_audit_cleanup(days number) as
  ver varchar2(100);
begin
  select version into ver from v$instance;

  if ver like '10%' or ver like '11%'
  then
    execute immediate 'delete from sys.aud$
                         where extract(day from
sys_extract_utc(systimestamp)-ntimestamp#) > ' || days ||
                        ' and sessionid not in (select audsid from
sys.V$SESSION)';

    execute immediate 'delete from sys.fga_log$
                         where extract(day from
sys_extract_utc(systimestamp)-ntimestamp#) > ' || days ||
                        ' and sessionid not in (select audsid from
sys.V$SESSION)';
  else
    if ver like '9%' then
      execute immediate 'delete from sys.aud$
                           where extract(day from
sys_extract_utc(systimestamp)-timestamp#) > ' || days ||
                          ' and sessionid not in (select audsid from
sys.V$SESSION)';

      execute immediate 'delete from sys.fga_log$
                           where extract(day from
sys_extract_utc(systimestamp)-timestamp#) > ' || days ||
                          ' and sessionid not in (select audsid from
sys.V$SESSION)';
    end if;
  end if;
```

```
end;
/

-- the parameter for no. of days is configurabl
-- change the value of "no_of_days" (current value = 7)
Declare
  ver varchar2(100);
  jobno binary_integer;
------
  no_of_days number := 7;
------
begin
  select version into ver from v$instance;

  if ver like '10%' or ver like '11%'
  then
    execute immediate
      'begin DBMS_SCHEDULER.CREATE_JOB (
                  JOB_NAME => ''AUDIT_CLEANUP'',
                  JOB_TYPE => ''PLSQL_BLOCK'',
                  JOB_ACTION => ''begin sys.source_audit_cleanup(days
=> ' || no_of_days ||'); end;'',
                  REPEAT_INTERVAL => ''FREQ=DAILY;INTERVAL=1'',
                  ENABLED => TRUE,
                  AUTO_DROP => FALSE,
                  COMMENTS => ''Cleaup Job Run Daily''); end;';
  else if ver like '9%' then
    execute immediate 'begin dbms_job.submit(job =>  :jobno,
                                             what => ''begin
sys.source_audit_cleanup('|| no_of_days ||'); end;'',
                                             interval => ''SYSDATE +
1''); end ;' using in out jobno;
    dbms_output.put_line('Job No ' || jobno);
    commit;
    end if;
  end if;
end;
/
```

Appendix B. Database Source Audit Settings

```
Rem
Rem Copyright (c) 2007, Oracle. All rights reserved.
Rem
Rem    DESCRIPTION
Rem      Secure configuration settings for the database include audit
REM      settings (enabled, with admin actions audited.
Rem

-- Turn on auditing options

Audit alter any table by access;

Audit create any table by access;

Audit drop any table by access;

Audit Create any procedure by access;

Audit Drop any procedure by access;

Audit Alter any procedure by access;

Audit Grant any privilege by access;

Audit grant any object privilege by access;

Audit grant any role by access;

Audit audit system by access;

Audit create external job by access;

Audit create any job by access;

Audit create any library by access;

Audit create public database link by access;

Audit exempt access policy by access;

Audit alter user by access;

Audit create user by access;

Audit role by access;

Audit create session by access;

Audit drop user by access;

Audit alter database by access;

Audit alter system by access;
```

```
Audit alter profile by access;

Audit drop profile by access;
```

ORACLE

Oracle Audit Vault Best Practices
November 2007
Author: Tammy Bednar
Contributions: Paul Needham, Vipul Shah

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com