EventTracker
Secure. Comply. Succeed.

Monitor Oracle Event Logs using EventTracker

Publication Date: Oct 23, 2013

EventTracker
8815 Centre Park Drive
Columbia MD 21045
www.eventtracker.com

Abstract

The purpose of this paper is to highlight the major advantages of employing EventTracker to consolidate and manage Oracle 10g and above log data. The paper introduces at a high level the major design concepts that enable EventTracker to process, store and allow users to gain actionable intelligence from the millions of critical events generated by Oracle.

Oracle event data contains a wealth of valuable information for Oracle DBAs and for security controls and compliance. Monitoring and managing Oracle event logs manually is tedious, time consuming and practically impossible for a large setup. This paper explains how effectively and efficiently these jobs can be performed using EventTracker.

The steps mentioned here for setting up of the Oracle audit trail is Operating System independent. Oracle 10g and above internally checks for Operating System type and forwards the syslog event logs accordingly.

Scope

The configurations detailed in this guide are consistent with **EventTracker Enterprise** version 7.X and later, and Oracle Database 10g and later.

Audience

Oracle Database users, who wish to forward auditing events to EventTracker Manager.

The information contained in this document represents the current view of Prism Microsystems Inc. on the issues discussed as of the date of publication. Because Prism Microsystems must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Prism Microsystems, and Prism Microsystems cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. Prism Microsystems MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AS TO THE INFORMATION IN THIS DOCUMENT.

The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.

Table of Contents

Oracle 10g Server Audit Setup

Auditing is a default feature of the Oracle server, and is disabled by default. However, it can be enabled by setting the AUDIT_TRAIL static parameter which has the following allowed values.

AUDIT_TRAIL= { none | os | db | db,extended | xml | xml,extended }

The following list provides a description of each setting:

- None or false - Auditing is disabled.
- db or true – Auditing is enabled, will audit records stored in the database audit trail (SYS.AUD$)
- db,extended – As db, but the SQL_BIND and SQL_TEXT columns are so populated.
- xml – Auditing is enabled, with all audit records stored as XML format OS files.
- xml,extended – As xml, but the SQL_BIND and SQL_TEXT columns are also populated.
- os- Auditing is enabled, with all audit records directed to the operating system's audit trail. In case of Windows Operating System the audit logs will be forwarded to windows event viewer, in case of Linux or Unix Operating System the audit logs will be forwarded to syslog.

Operating System Audit Trail Enabling

The Operating System used for auditing Oracle event logs are

- Audit Trail for Windows Operating System
- Audit Trail for Linux/Solaris Operating System

Audit Trail for the Windows Operating System

Next enable AUDIT_TRAIL for the Operating System. In the case of a Windows Operating System, it will forward all the audit events to the Windows event viewer.

If the OS audit trail is not enabled, Oracle will continue to write default actions to the OS audit trail:

- instance startup
- instance shutdown
- connections to the database as SYSOPER or SYSDBA

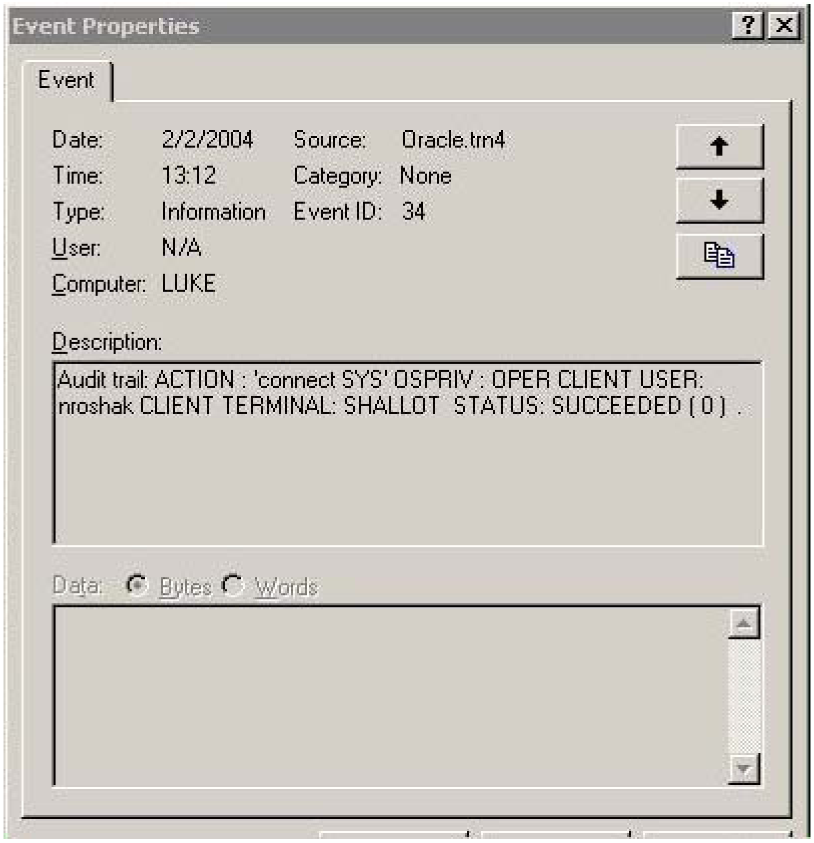
To enable auditing and direct audit records to the os audit trail, do the following:

```
SQL> ALTER SYSTEM SET audit_trail=os SCOPE=spfile;
System altered.
SQL> SHUTDOWN
Database closed.
Database dismounted.
ORACLE instance shutdown.
SQL> STARTUP
ORACLE instance started.
Total System Global Area 289406976 bytes
Fixed Size 1248600 bytes
Variable Size 71303848 bytes
Database Buffers 213909504 bytes
Redo Buffers 2945024 bytes
Database mounted.
Database opened.
```

It is compulsory either to restart the Oracle server after changing any audit options or set AUDIT_TRAIL = OS parameter in the INIT.ORA file located by default under the %ORACLE_HOME%database path.

Operating System Audit Trail Illustrated for Windows

1. Select the **Start** button, select **Control Panel,** and then select **Administrative Tools**.
2. Select **Windows Event Viewer,** select **Application Logs**.
3. Go to **View Tools**, and click **Filter** option **Oracle.your_SID** as your Event Source.
   This will list all Oracle generated events. Events generated using Create User Statement will look like the example below.



Date: 2/2/2004 Source: Oracle.trn4
Time: 13:12 Category: None
Type: Information Event ID: 34
User: N/A
Computer: LUKE

Description:

Audit trail: ACTION : 'connect SYS' OSPRIV : OPER CLIENT USER: nroshak CLIENT TERMINAL: SHALLOT STATUS: SUCCEEDED (0) .

Figure 1: Example of OS Audit Trail

Audit DBA Activities

DBAs are normally exempted from fine-grained access to auditing. To turn on audit for DBA activities follow the steps below:

SQL> ALTER SYSTEM SET audit_sys_operations = TRUE SCOPE=SPFILE;

System altered.

SQL> SHUTDOWN

```
Database closed.

Database dismounted.

ORACLE instance shut down.

SQL> STARTUP

ORACLE instance started.

Total System Global Area 289406976 bytes

Fixed Size 1248600 bytes

Variable Size 71303848 bytes

Database Buffers 213909504 bytes

Redo Buffers 2945024 bytes

Database mounted.

Database opened.
```

Oracle Audit Options

There are two types of Audit options

- BY SESSION: Using this option only one audit record is inserted for one session regardless of number of times the statement is executed. BY SESSION is the default.
- BY ACCESS: Using this option one audit record is inserted each time the statement is executed.

There are three levels of audit options

1. Statement
    - Audits on the type of SQL statement used, such as any SQL statement on a table (which records each CREATE, TRUNCATE and DROP TABLE statement)
    - Ex. AUDIT SELECT BY SCOTT audits all select statements performed by SCOTT
    - Ex. AUDIT SELECT TABLE, UPDATE TABLE BY SCOTT, MICKEY; audits all the select, update statements by SCOTT and MICKEY.

2. Object

- Audits specific statements on specific objects such as ALTER TABLE on the EMP table
- Ex. AUDIT SELECT ON scott.emp; audits selection statements of all users on scott.emp table.
- Ex. AUDIT SELECT ON scott.emp WHENEVER NOT SUCCESSFUL; audits selection statements of all users on scott.emp table whenever not successful (only failure conditions).
- Ex. AUDIT SELECT ON scott.emp WHENEVER SUCCESSFUL; audits selection statements of all users on scott.emp table whenever successful (only successful conditions)
- Ex. AUDIT ALL ON scott.emp; audits all the conditions by the entire user on scott.emp table

3. Privilege

A. Audits use of a particular system privilege such as CREATE TABLE

B. Ex. AUDIT GRANT SEQUENCE; audits any statements of the type:
1) GRANT privilege ON sequence
2) REVOKE privilege ON sequence

C. Ex. AUDIT EXECUTE PROCEDURE; audits
1) CALL of any procedure

D. Ex. AUDIT SELECT TABLE; audits
1) SELECT FROM table/view/materialized view;

E. Ex. AUDIT INDEX audits any statement of the type:
1) CREATE INDEX
2) ALTER INDEX
3) DROP INDEX

F. Ex. AUDIT NOT EXISTS audits all SQL statements that fail because an object doesn’t exist

G. Ex. AUDIT SYSTEM AUDIT audits all AUDIT/NOAUDIT statements

H. E.g. AUDIT SESSION audits logon/logoff

Oracle Audit Action Events

Generally all audit events generated by Oracle use unique action numbers in the range of 0 to 211. Below is the list of action numbers and their respective action names:

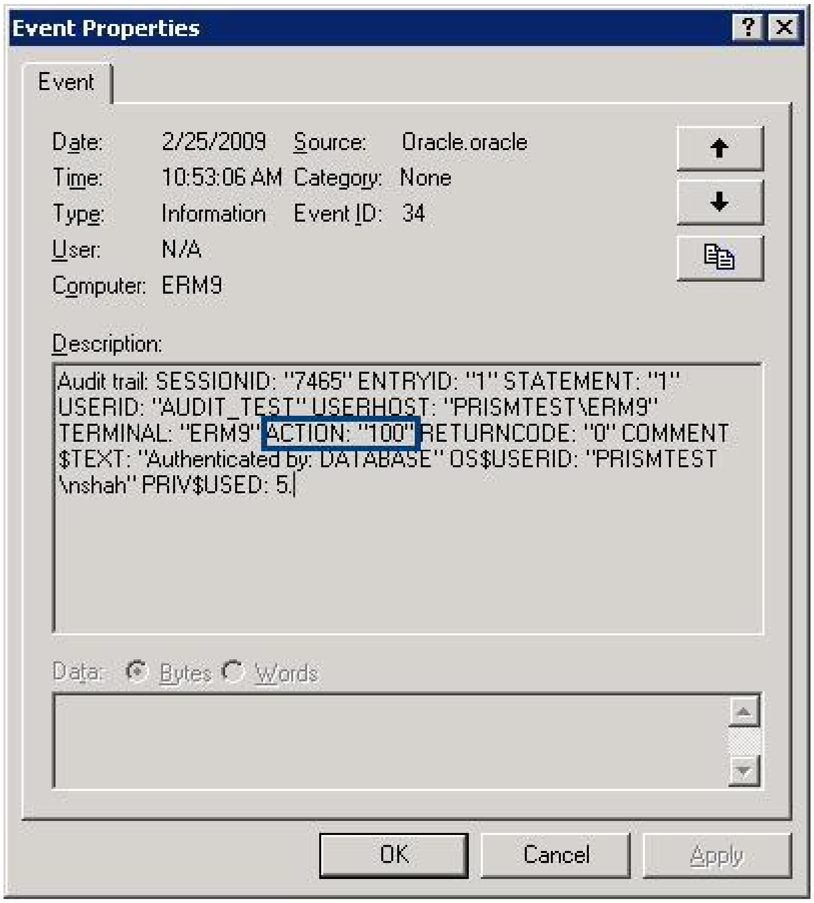
Action ID	Action Name
0	UNKNOWN
1	CREATE TABLE
2	INSERT
3	SELECT
4	CREATE CLUSTER
5	ALTER CLUSTER
6	UPDATE
7	DELETE
8	DROP CLUSTER
9	CREATE INDEX
10	DROP INDEX
11	ALTER INDEX
12	DROP TABLE
13	CREATE SEQUENCE
14	ALTER SEQUENCE
15	ALTER TABLE
16	DROP SEQUENCE
17	GRANT OBJECT
18	REVOKE OBJECT
19	CREATE SYNONYM
20	DROP SYNONYM
21	CREATE VIEW
22	DROP VIEW
23	VALIDATE INDEX
24	CREATE PROCEDURE
25	ALTER PROCEDURE
26	LOCK
27	NO-OP
28	RENAME
29	COMMENT
30	AUDIT OBJECT
31	NOAUDIT OBJECT
32	CREATE DATABASE LINK
33	DROP DATABASE LINK
34	CREATE DATABASE
35	ALTER DATABASE
36	CREATE ROLLBACK SEG
37	ALTER ROLLBACK SEG
38	DROP ROLLBACK SEG

39	CREATE TABLESPACE
40	ALTER TABLESPACE
41	DROP TABLESPACE
42	ALTER SESSION
43	ALTER USER
44	COMMIT
45	ROLLBACK
46	SAVEPOINT
47	PL/SQL EXECUTE
48	SET TRANSACTION
49	ALTER SYSTEM
50	EXPLAIN
51	CREATE USER
52	CREATE ROLE
53	DROP USER
54	DROP ROLE
55	SET ROLE
56	CREATE SCHEMA
57	CREATE CONTROL FILE
59	CREATE TRIGGER
60	ALTER TRIGGER
61	DROP TRIGGER
62	ANALYZE TABLE
63	ANALYZE INDEX
64	ANALYZE CLUSTER
65	CREATE PROFILE
66	DROP PROFILE
67	ALTER PROFILE
68	DROP PROCEDURE
70	ALTER RESOURCE COST
71	CREATE MATERIALIZED VIEW LOG
72	ALTER MATERIALIZED VIEW LOG
73	DROP MATERIALIZED VIEW LOG
74	CREATE MATERIALIZED VIEW
75	ALTER MATERIALIZED VIEW
76	DROP MATERIALIZED VIEW
77	CREATE TYPE
78	DROP TYPE
79	ALTER ROLE
80	ALTER TYPE
81	CREATE TYPE BODY
82	ALTER TYPE BODY
83	DROP TYPE BODY
84	DROP LIBRARY
85	TRUNCATE TABLE
86	TRUNCATE CLUSTER

91	CREATE FUNCTION
92	ALTER FUNCTION
93	DROP FUNCTION
94	CREATE PACKAGE
95	ALTER PACKAGE
96	DROP PACKAGE
97	CREATE PACKAGE BODY
98	ALTER PACKAGE BODY
99	DROP PACKAGE BODY
100	LOGON
101	LOGOFF
102	LOGOFF BY CLEANUP
103	SESSION REC
104	SYSTEM AUDIT
105	SYSTEM NOAUDIT
106	AUDIT DEFAULT
107	NOAUDIT DEFAULT
108	SYSTEM GRANT
109	SYSTEM REVOKE
110	CREATE PUBLIC SYNONYM
111	DROP PUBLIC SYNONYM
112	CREATE PUBLIC DATABASE LINK
113	DROP PUBLIC DATABASE LINK
114	GRANT ROLE
115	REVOKE ROLE
116	EXECUTE PROCEDURE
117	USER COMMENT
118	ENABLE TRIGGER
119	DISABLE TRIGGER
120	ENABLE ALL TRIGGER
121	DISABLE ALL TRIGGER
122	NETWORK ERROR
123	EXECUTE TYPE
128	FLASHBACK
129	CREATE SESSION
157	CREATE DIRECTORY
158	DROP DIRECTORY
159	CREATE LIBRARY
160	CREATE JAVA
161	ALTER JAVA
162	DROP JAVA
163	CREATE OPERATOR
164	CREATE INDEXTYPE
165	DROP INDEXTYPE
167	DROP OPERATOR
168	ASSOCIATE STATISTICS

169	DISASSOCIATE STATISTICS
170	CALL METHOD
171	CREATE SUMMARY
172	ALTER SUMMARY
173	DROP SUMMARY
174	CREATE DIMENSION
175	ALTER DIMENSION
176	DROP DIMENSION
177	CREATE CONTEXT
178	DROP CONTEXT
179	ALTER OUTLINE
180	CREATE OUTLINE
181	DROP OUTLINE
182	UPDATE INDEXES
183	ALTER OPERATOR
197	PURGE USER_RECYCLEBIN
198	PURGE DBA_RECYCLEBIN
199	PURGE TABLESPACE
200	PURGE TABLE
201	PURGE INDEX
202	UNDROP OBJECT
204	FLASHBACK DATABASE
205	FLASHBACK TABLE
206	CREATE RESTORE POINT
207	DROP RESTORE POINT
208	PROXY AUTHENTICATION ONLY
209	DECLARE REWRITE EQUIVALENCE
210	ALTER REWRITE EQUIVALENCE
211	DROP REWRITE EQUIVALENCE

Oracle Audit Action Codes Illustrated

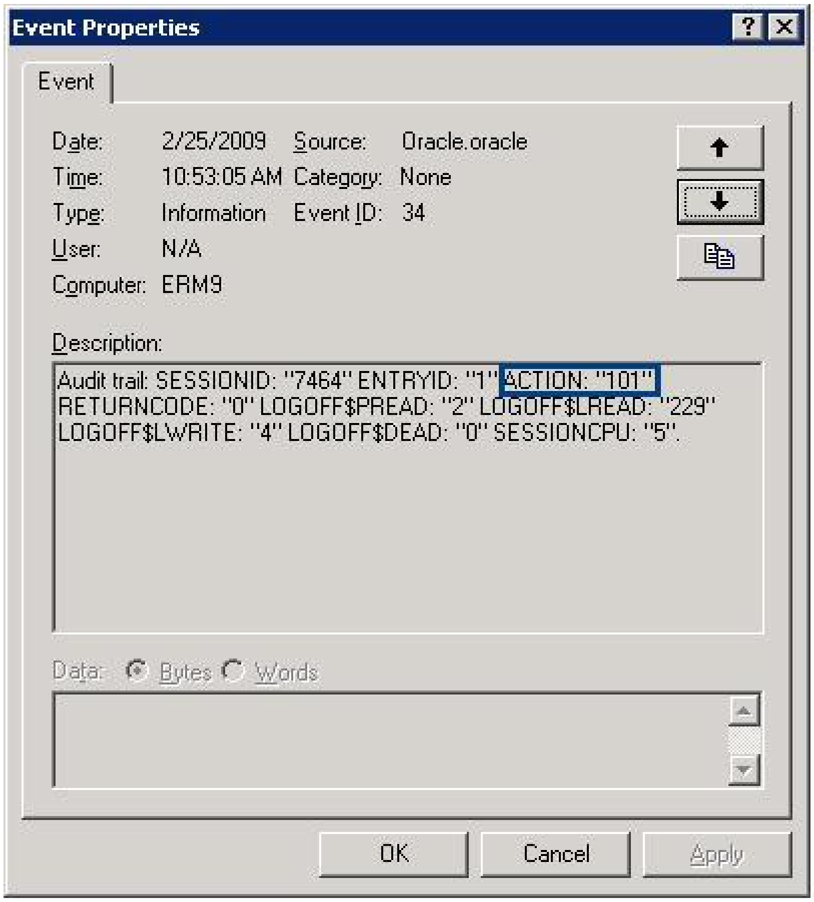


Date: 2/25/2009 Source: Oracle.oracle
Time: 10:53:06 AM Category: None
Type: Information Event ID: 34
User: N/A
Computer: ERM9

Description:

```
Audit trail: SESSIONID: "7465" ENTRYID: "1" STATEMENT: "1"
USERID: "AUDIT_TEST" USERHOST: "PRISMTEST\ERM9"
TERMINAL: "ERM9" ACTION: "100" RETURNCODE: "0" COMMENT
$TEXT: "Authenticated by: DATABASE" OS$USERID: "PRISMTEST
\nshah" PRIV$USED: 5.
```

Figure 2: The Action Code for a Logon event is 100

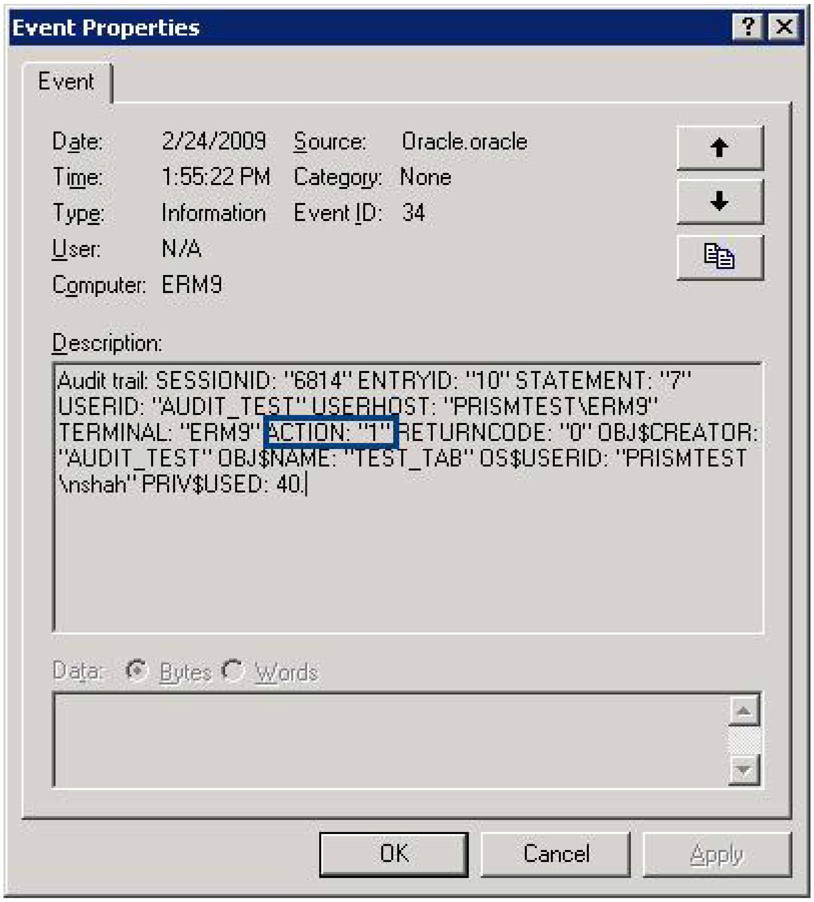


Date: 2/25/2009 Source: Oracle.oracle
Time: 10:53:05 AM Category: None
Type: Information Event ID: 34
User: N/A
Computer: ERM9

Description:

```
Audit trail: SESSIONID: "7464" ENTRYID: "1" ACTION: "101"
RETURNCODE: "0" LOGOFF$PREAD: "2" LOGOFF$LREAD: "229"
LOGOFF$LWRITE: "4" LOGOFF$DEAD: "0" SESSIONCPU: "5".
```

Figure 3: The Action Code for a Logoff event is 101



Date: 2/24/2009 Source: Oracle.oracle
Time: 1:55:22 PM Category: None
Type: Information Event ID: 34
User: N/A
Computer: ERM9

Description:

```
Audit trail: SESSIONID: "6814" ENTRYID: "10" STATEMENT: "7"
USERID: "AUDIT_TEST" USERHOST: "PRISMTEST\ERM9"
TERMINAL: "ERM9" ACTION: "1" RETURNCODE: "0" OBJ$CREATOR:
"AUDIT_TEST" OBJ$NAME: "TEST_TAB" OS$USERID: "PRISMTEST
\nshah" PRIV$USED: 40.
```

Data: ● Bytes ○ Words

Figure 4: The Action Code for a Create Table event is 1

Monitor, Alert and Report Oracle Audit Logs with EventTracker

Once Oracle 10g is configured to trail audit logs into the Operating System log, EventTracker provides support for efficiently monitoring, alerting and reporting of Oracle Audit Logs.

Monitor

EventTracker monitors all events generated by an Oracle audit trail. DBA can monitor specific groups of events like logon failure events, table deletion (success and failure) events, User creation (success and failure) events, etc.

Alert

EventTracker can alert DBAs on critical events such as login failures on Database, Deletion of a table, Deletion of User, Creation of User. These alerts can be received via email, or SNMP traps, and as RSS feeds.

Report

EventTracker provides an exclusive reporting tool to generate requirement specific reports. Manual logging makes it difficult to retrieve the list of logon failures or users created between certain time intervals. Below are sample reports created by EventTracker specific to Oracle Audit Trail logs.

User Logon Success Report

Summary Report(s)

Computer	Total Event Occurred	Event ID (Total count)
ERM9	85	34 (85)

Detail Report

Computer	Log Time	User name	Privilege Used
ERM9	2/24/2009 14:01	DBSNMP	5
ERM9	2/24/2009 10:53	SCOTT	5
ERM9	2/24/2009 13:41	DBSNMP	5
ERM9	2/24/2009 13:47	SYSMAN	5
ERM9	2/24/2009 11:25	SCOTT	5
ERM9	2/24/2009 12:40	MICHALE	5
ERM9	2/24/2009 12:44	SCOTT	5
ERM9	2/24/2009 12:50	NIKUNJ	5
ERM9	2/24/2009 12:57	DBSNMP	5
ERM9	2/24/2009 14:36	DBSNMP	5
ERM9	2/24/2009 13:06	SYSMAN	5

Oracle User Logon Failure Report

Summary Report(s)

Computer	Total Event Occurred	Event ID (Total count)
ERM9	6	34 (6)

Detail Report

Computer	Log Time	User name	Privilege Used
ERM9	2/23/2009 12:44	SCOTT	1017
ERM9	2/24/2009 12:49	NIKUNJ	1045
ERM9	2/24/2009 12:43	SCOTT	1017
ERM9	2/24/2009 12:40	MICHALE	1017
ERM9	2/24/2009 11:25	SCOTT	1017

Oracle - Create User Success Report

Summary Report(s)

Computer	Total Event Occurred	Event ID (Total count)
ERM9	2	34 (2)

Detail Report

Computer	Log Time	Created By	Created User Name
ERM9	2/24/2009 12:49	SYSDBA	NIKUNJ
ERM9	2/24/2009 12:18	SYSDBA	MICHALE

Oracle - Create User Failure Report

Summary Report(s)

Computer	Total Event Occurred	Event ID (Total count)
ERM9	2	34 (2)

Detail Report

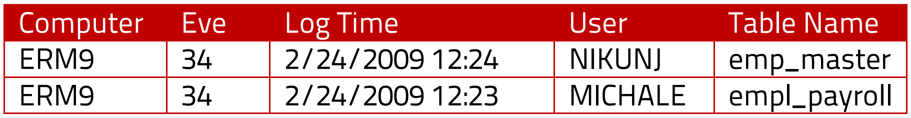
Computer	Log Time	Created By	Created User Name	Return Code	Status
ERM9	2/24/2009 12:24	SYSDBA	NIKUNJ		1920
ERM9	2/24/2009 12:23	SYSDBA	MICHALE	1031	

Oracle - Create Table Success Report

Summary Report(s)

Computer	Total Event Occurred	Event ID (Total count)
ERM9	2	34 (2)

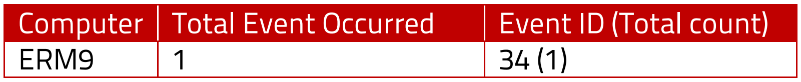
Detail Report



Computer	Eve	Log Time	User	Table Name
ERM9	34	2/24/2009 12:24	NIKUNJ	emp_master
ERM9	34	2/24/2009 12:23	MICHALE	empl_payroll

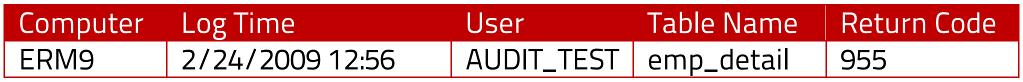
Oracle - Create Table Failure Report

Summary Report(s)



Computer	Total Event Occurred	Event ID (Total count)
ERM9	1	34 (1)

Detail Report



Computer	Log Time	User	Table Name	Return Code
ERM9	2/24/2009 12:56	AUDIT_TEST	emp_detail	955

Audit Trail for Linux/Solaris Operating System

Pre-requisites for Linux/Solaris

Before you begin

- EventTracker version 7.X and later must be installed
- Solaris 10/Linux Operating system must be installed
- Oracle Database 10g and above must be installed & configured
- Oracle database connectivity must be there

Configurations for Linux/Solaris

Oracle 10g and above versions support sending the Oracle Audit Logs to SYSLOG. EventTracker for Oracle Audit Trails can receive syslog from Oracle.

1. Login to Oracle user terminal.
   ```
   $ ORACLE_SID=DB11G(Database name)
   $ Export ORACLE_SID
   ```

2. Connect to a database instance as sysdba user.
   ```
   $ sqlplus / as sysdba
   SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 9 23:42:07 2013

   Copyright (c) 1982, 2009, Oracle.  All rights reserved.

   Connected to:
   Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
   With the Partitioning, OLAP, Data Mining and Real Application Testing options
   SQL>
   ```

3. Set audit trail to OS.
   ```
   SQL> ALTER SYSTEM SET audit_trail=os SCOPE=SPFILE;
   System altered.
   ```

4. Enable auditing for system operations.
   ```
   SQL> ALTER SYSTEM SET audit_sys_operations = TRUE SCOPE=SPFILE;
   System altered.
   ```

5. Set syslog facility and severity.

   ```
   SQL> ALTER SYSTEM SET audit_syslog_level='local1.info' SCOPE=SPFILE;
   System altered.
   ```

6. Generate **'pfile'** from **'spfile'**.

   ```
   SQL> create pfile='/export/home/u01/app/oracle/admin/DB11G/pfile/init.ora.9920134550' from spfile;
   File created.
   ```

   NOTE:

   Make sure you mention a location for the pfile else it will overwrite the init.ora.9920134550 file under /export/home/u01/app/oracle/admin/DB11G/pfile

7. Verify that pfile is created on the mentioned location.

8. After confirmation, shutdown the instance

   ```
   SQL> shutdown immediate
   Database closed.
   Database dismounted.
   ORACLE instance shut down.
   ```

9. Startup the database.

   ```
   SQL> startup
   ORACLE instance started.

   Total System Global Area 1720328192 bytes
   Fixed Size                 2211728 bytes
   Variable Size           1040187504 bytes
   Database Buffers         671088640 bytes
   Redo Buffers               6840320 bytes
   Database mounted.
   Database opened.
   ```

   - If it's not starting then start using pfile

     ```
     SQL> startup pfile= '/export/home/u01/app/oracle/admin/DB11G/pfile/init.ora.9920134550'
     ```

10. Generate 'spfile' from 'pfile'.

    ```
    SQL> create spfile='/export/home/u01/app/oracle/product/11.2.0.2/db_1/dbs/spfileDB11G.ora' from pfile='/export/home/u01/app/oracle/admin/DB11G/pfile/init.ora.9920134550';
    ```

File created.

11. Edit /etc/syslog config file to forward local1.info to EventTracker.
    In case of syslog, edit the /etc/syslog.conf file and set the following configurations given below.

    The first entry is for the local syslog.

    ```
    #Save oracle rdbms audit trail to oracle_audit.log
    local1.info                    /var/log/oracle/oracle_audit.log
    ```

    The second entry sends it to a remote server (EventTracker Manager):

    ```
    #Send oracle rdbms audit trail to remote syslog server
    local1.info                    @192.168.1.45
    ```

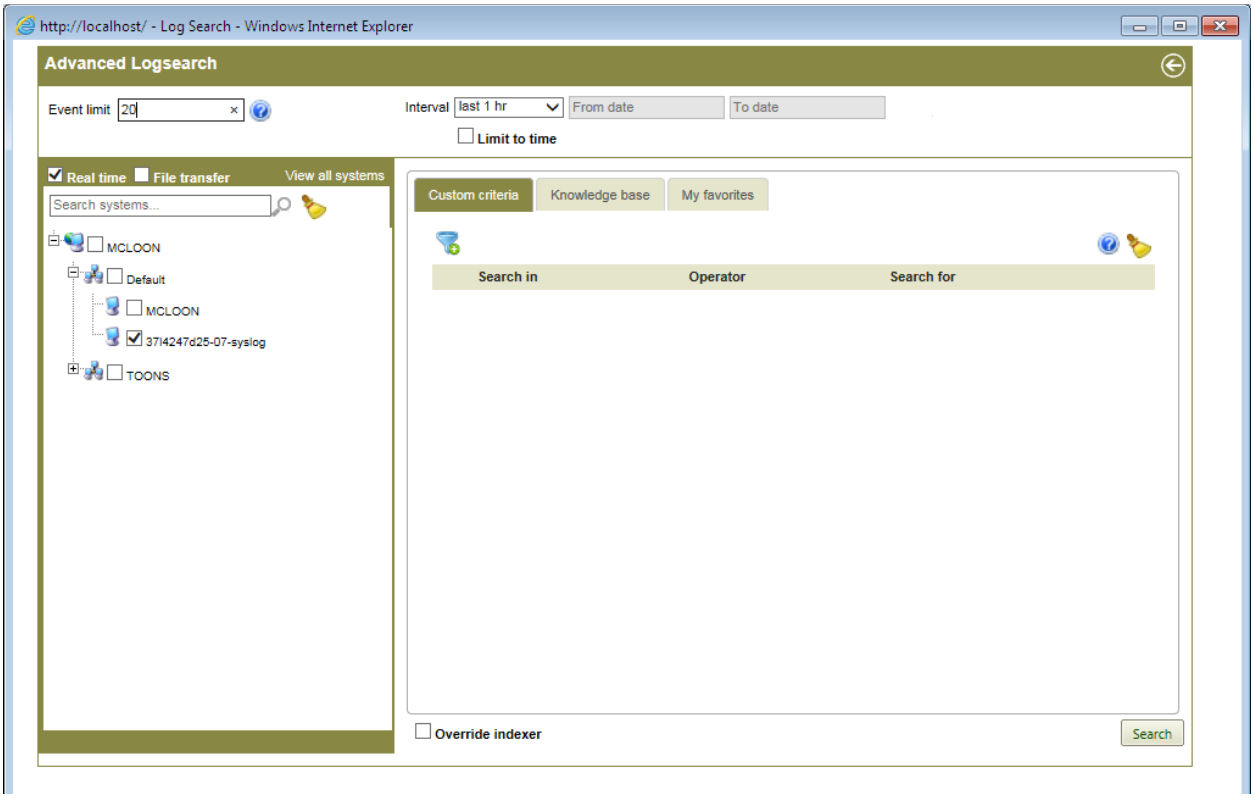
12. Restart you syslog service.

    ```
    #svcadm restart system/system-log
    ```

13. Verify in EventTracker, if logs are generated.
    The details are mentioned below.

Verify Oracle Event Logs in EventTracker

1. Logon to EventTracker Enterprise.
2. Select the **Search** menu, and then select **Advance Search**.
   Log Search window displays.



Event limit 20

Interval last 1 hr

□ Limit to time

☑ Real time □ File transfer

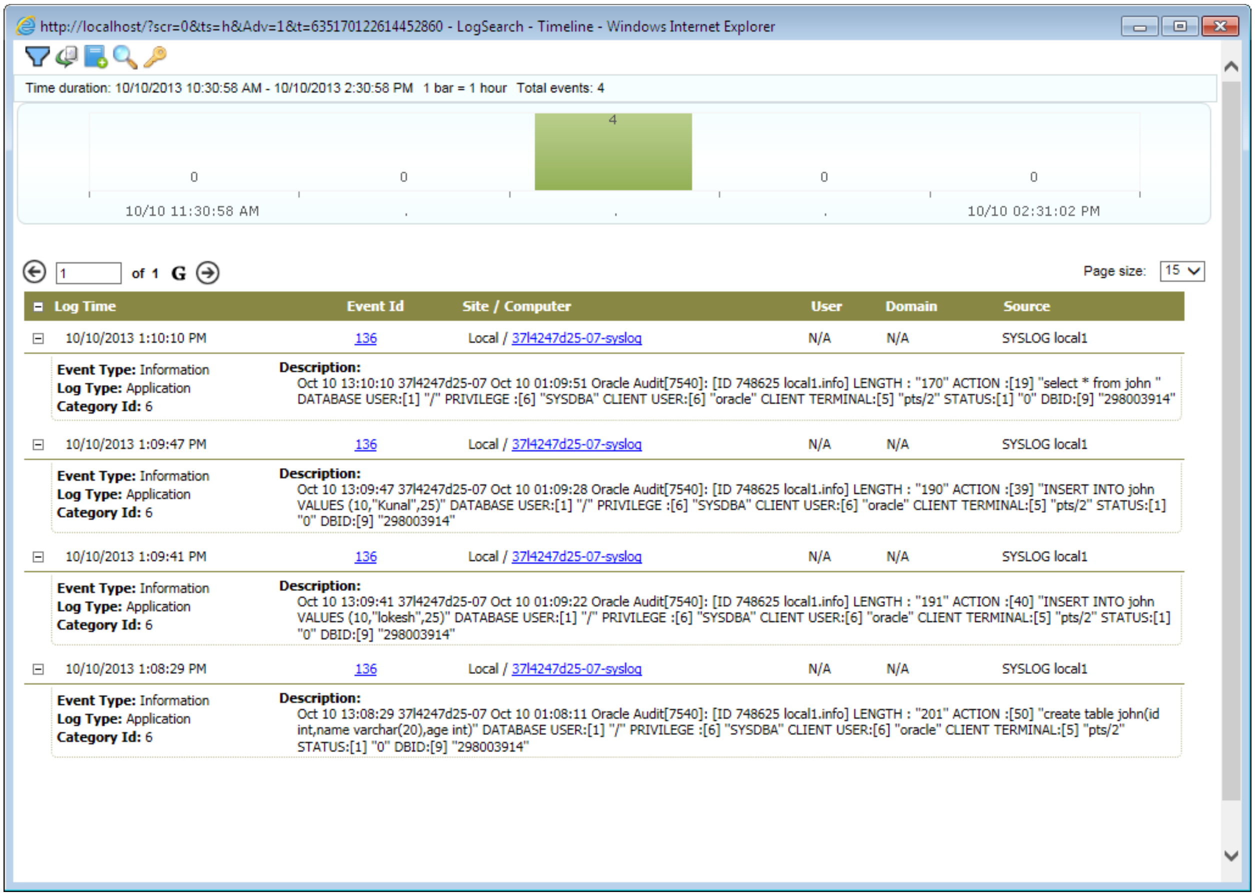
☑ 37l4247d25-07-syslog

Custom criteria

□ Override indexer

3. Select the system in which logs are generated and also the **Interval**.
4. Select the **Search** button.

   The resultant output displays.


http://localhost/?scr=0&ts=h&Adv=1&t=635170122614452860 - LogSearch - Timeline - Windows Internet Explorer
4
0
0
0
0
10/10 11:30:58 AM
10/10 02:31:02 PM
1 of 1 G
Page size: 15

Time duration: 10/10/2013 10:30:58 AM - 10/10/2013 2:30:58 PM 1 bar = 1 hour Total events: 4

Log Time	Event Id	Site / Computer	User	Domain	Source
10/10/2013 1:10:10 PM	136	Local / 37l4247d25-07-syslog	N/A	N/A	SYSLOG local1

Event Type: Information
Log Type: Application
Category Id: 6

Description:
Oct 10 13:10:10 37l4247d25-07 Oct 10 01:09:51 Oracle Audit[7540]: [ID 748625 local1.info] LENGTH : "170" ACTION :[19] "select * from john " DATABASE USER:[1] "/" PRIVILEGE :[6] "SYSDBA" CLIENT USER:[6] "oracle" CLIENT TERMINAL:[5] "pts/2" STATUS:[1] "0" DBID:[9] "298003914"

10/10/2013 1:09:47 PM	136	Local / 37l4247d25-07-syslog	N/A	N/A	SYSLOG local1

Event Type: Information
Log Type: Application
Category Id: 6

Description:
Oct 10 13:09:47 37l4247d25-07 Oct 10 01:09:28 Oracle Audit[7540]: [ID 748625 local1.info] LENGTH : "190" ACTION :[39] "INSERT INTO john VALUES (10,"Kunal",25)" DATABASE USER:[1] "/" PRIVILEGE :[6] "SYSDBA" CLIENT USER:[6] "oracle" CLIENT TERMINAL:[5] "pts/2" STATUS:[1] "0" DBID:[9] "298003914"

10/10/2013 1:09:41 PM	136	Local / 37l4247d25-07-syslog	N/A	N/A	SYSLOG local1

Event Type: Information
Log Type: Application
Category Id: 6

Description:
Oct 10 13:09:41 37l4247d25-07 Oct 10 01:09:22 Oracle Audit[7540]: [ID 748625 local1.info] LENGTH : "191" ACTION :[40] "INSERT INTO john VALUES (10,"lokesh",25)" DATABASE USER:[1] "/" PRIVILEGE :[6] "SYSDBA" CLIENT USER:[6] "oracle" CLIENT TERMINAL:[5] "pts/2" STATUS:[1] "0" DBID:[9] "298003914"

10/10/2013 1:08:29 PM	136	Local / 37l4247d25-07-syslog	N/A	N/A	SYSLOG local1

Event Type: Information
Log Type: Application
Category Id: 6

Description:
Oct 10 13:08:29 37l4247d25-07 Oct 10 01:08:11 Oracle Audit[7540]: [ID 748625 local1.info] LENGTH : "201" ACTION :[50] "create table john(id int,name varchar(20),age int)" DATABASE USER:[1] "/" PRIVILEGE :[6] "SYSDBA" CLIENT USER:[6] "oracle" CLIENT TERMINAL:[5] "pts/2" STATUS:[1] "0" DBID:[9] "298003914"

About EventTracker

The EventTracker solution is a scalable, enterprise-class Security Information and Event Management (SIEM) solution for Windows systems, Syslog/Syslog NG (UNIX and many networking devices), SNMP V1/2, legacy systems, applications and databases. EventTracker enables 'defense in depth", where log data is automatically collected, correlated and analyzed from the perimeter security devices down to the applications and databases.

To prevent security breaches, event log data becomes most useful when interpreted in near real time and in context. Context is vitally important because often the critical indications of impending problems and security violations can only be learned by watching patterns of events across multiple systems. Complex rules can be run on the event stream to detect signs of such a breach. EventTracker also provides real-time alerting capability in the form of an email, page or SNMP message to proactively alert security personnel to an impending security breach.

The original log data is also securely stored in a highly compressed event repository for compliance purposes and later forensic analysis. For compliance, EventTracker provides a powerful reporting interface, scheduled or on-demand report generation, automated compliance workflows that prove to auditors that reports are being reviewed and many other features. With prebuilt auditor grade reports included for most of the compliance standards (FISMA, HIPAA, SOX, GLBA and more); EventTracker represents a compliance solution that is second to none. EventTracker also provides advanced forensic capability where all the stored logs can be quickly searched through a powerful Google-like search interface to perform quick problem determination.

EventTracker lets users completely meet the logging requirements specified in NIST SP 800-92 Guide To Computer Security Log Management, and additionally provides Host Based Intrusion Detection, Change Monitoring and USB activity tracking on Windows systems, all in an off the shelf, affordable, software solution.

EventTracker provides the following benefits:

- A highly scalable, component-based architecture that consolidates all Windows, SNMP V1/V2, legacy platforms, Syslog received from routers, switches, firewalls, critical UNIX servers (Red Hat Linux, Solaris, AIX etc), Solaris BSM, workstations and various other SYSLOG generating devices.

- Automated archival mechanism that stores activities over an extended period to meet auditing requirements. The complete log is stored in a highly compressed (>90%), secured (Sealed with SHA1 – checksum) archive that is limited only by the amount of available disk storage.
- Real-time monitoring and parsing of all logs to analyze user activities such as logon failures and failed attempts to access restricted information.
- Alerting interface that generates custom alert actions via email, pager, console message, etc.
- Event correlation modules to constantly monitor for malicious hacking activity. In conjunction with alerts, this is used to inform network security officers and security administrators in real time. This helps minimize the impact of breaches.
- Various types of network activity reports, which can be scheduled or generated as required for any investigation or meeting audit compliances
- Host-based Intrusion Detection (HIDS).
- Role-based, secure event and reporting console for data analysis.
- Change Monitoring on Windows machines
- USB Tracking, including restricted use, insert/removal recording, and a complete audit trail of all files copied to the removable device.
- Built-in compliance workflows to allow inspection and annotation of the generated reports.

About Prism Microsystems

Prism Microsystems, Inc. delivers business-critical solutions to consolidate, correlate and detect changes that could impact the performance, availability and security of your IT infrastructure. With a proven history of innovation and leadership, Prism provides easy-to-deploy products and solutions for integrated Security Management, Change Management and Intrusion Detection EventTracker, Prism's market leading enterprise log management solution, enable commercial enterprises, educational institutions and government organizations to increase the security of their environments and reduce risk to their enterprise. Customers span multiple sectors including financial, communications, scientific, healthcare, banking and consulting.

Prism Microsystems was formed in 1999 and is a privately held corporation with corporate headquarters in the Baltimore-Washington high tech corridor. Research and development facilities are located in both Maryland and India. These facilities have been independently appraised in accordance with the Software Engineering Institute's Appraisal Framework and were deemed to meet the goals of SEI Level 3 for CMM. For additional information, please visit http://www.eventtracker.com/.

.