INTEGRIGY

WHITE PAPER

Oracle PeopleSoft
Guide to Auditing and Logging

APRIL 2017

GUIDE TO AUDITING AND LOGGING IN ORACLE PEOPLESOFT

Version 1.0 – March 2017 - created
Version 2.0 – April 2017 – rewrite to focus on FGA

Authors: Mike Miller, CISSP, CISSP-ISSMP, CCSK

If you have any questions, comments, or suggestions regarding this document, please send them via e-mail to info@integrigy.com.

Table of Contents

Overview **4**

Integrigy's Log and Audit Framework for PeopleSoft **6**

Framework Approach 7

Log and Audit Functionality **10**

What Is a Log? 10
Operating system Logging 10
Oracle Database 10
PeopleSoft 12

Integrigy Framework – Level 1 ..... **16**

Database Auditing 16
PeopleSoft Auditing 19
Recommended Monitoring and Alerts 27
Consider Oracle Database Vault (Optional) 30

Integrigy Framework – Level 2 ..... **31**

Implement Centralized Logging Solution 31
Redirect Database Logs to Centralized Logging 31
Transition Level 1 Alerts and Build Additional Level 2 Alerts 32

Integrigy Framework – Level 3 ..... **34**

Additional Database and Application Logs 34
Secure Sensitive Data 39

Appendix A – Recommendations for PeopleSoft Auditing **42**

Appendix B – PeopleSoft Triggers and Shadow Audit Tables **44**

Appendix C – Useful SQL **57**

References **59**

General 59

About Integrigy **60**

OVERVIEW

Most clients do not fully take advantage of PeopleSoft's auditing and logging features. These features are sophisticated and can satisfy most organization's compliance and security requirements.

The default PeopleSoft installation only provides a basic set of logging functionality. In Integrigy's experience, the implementation of database and application logging seldom exceeds meeting the needs of basic debugging. Most organizations do not know where to start or how to leverage the built-in auditing and logging features to satisfy their compliance and security requirements.

Even organizations already using centralized logging or Security Incident and Event Management (SIEM) solutions, while being more advanced in the Common Maturity Model (CMM), in Integrigy's experience are commonly challenged by PeopleSoft's auditing and logging features and functionality.

This guide presents Integrigy's framework for auditing and logging in PeopleSoft. This framework is a direct result of Integrigy's consulting experience and will be equally useful to both those wanting to improve their capabilities as well as those just starting to implement logging and auditing. Our goal is to provide a clear explanation of the native auditing and logging features available, present an approach and strategy for using these features and straightforward instructions to implement the approach.

Integrigy's framework is also specifically designed to help clients meet compliance and security standards such as Sarbanes-Oxley (SOX), Payment Card Industry (PCI), FISMA, and HIPAA. The foundation of the Framework is PCI DSS requirement 10.2.

To make it easy for clients to implement, the Framework has three maturity levels – which level a client starts at depends on the infrastructure and policies already in place.

The three levels are:

- **Level 1** – Enable baseline auditing and logging for application & database security sensitive events and implement security monitoring and auditing alerts
- **Level 2** – Send audit and log data to a centralized logging solution outside the Oracle Database and PeopleSoft
- **Level 3** – Extend logging to include functional logging and more complex alerting and monitoring and to secure sensitive data

Recommended Technical Approach

PeopleSoft does offer a database agnostic auditing solution that utilizes database triggers and shadow tables. This option is documented in Appendix B, but is not recommended primarily due to its overall limited efficiency due to its performance overhead and complexity to implement. For most clients, Fine Grained Auditing (FGA), free with the enterprise edition of the Oracle RDBMS, will more than meet their requirements for performance, compliance, and security and, for this reason, is the foundation of Integrigy's Audit Framework.

Versions, Audience and How to Read This Paper

The Framework was designed for PeopleTools 8.5x and version 9 of the Applications, however, the recommended use of Fine Grained Auditing (FGA) should work for Oracle RDBMS clients using lower versions of PeopleTools and/or the Applications.

The intended audience of this paper are PeopleSoft DBAs, application administrators, IT security staff, and internal audit staff. A working technical knowledge of PeopleSoft and Oracle Databases is recommended.

The section reviewing the logging functionality available in PeopleSoft and/or the Oracle Database may be skipped if the material is already familiar. Internal audit and IT security staff may find it useful to proceed directly to the presentation of Integrigy's Security Monitoring and Audit Framework

INTEGRIGY'S LOG AND AUDIT FRAMEWORK FOR PEOPLESOFT

The framework is a result of Integrigy's consulting experience and is based on compliance and security standards such as Payment Card Industry (PCI-DSS), Sarbanes-Oxley (SOX), IT Security (ISO 27001), FISMA (NIST 800-53), and HIPAA.

The foundation of the Framework is the set of security events and actions that should be audited and logged in all PeopleSoft implementations. These security events and actions are derived from and mapped back to key compliance and security standards that most organizations have to comply with. We view these security events and actions as the core set, and most organizations will need to expand these events and actions to address specific compliance and security requirements, such as functional or change management requirements.

Table 1 presents the core set of audits that, if implemented, will serve as a foundation for more advanced security analytics. Implementing these audits will go a long way toward meeting logging and auditing requirements for most compliance and security standards like PCI requirement 10.2. The numbering scheme used in Table 1 will be referenced throughout the document.

Table 1 – Foundation Events for Logging and Security Framework

Security Events and Actions	PCI DSS 10.2	SOX (COBIT)	HIPAA (NIST 800-66)	IT Security (ISO 27001)	FISMA (NIST 800-53)
E1 - Login	10.2.5	A12.3 DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E2 - Logoff	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E3 - Unsuccessful login	10.2.4	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1 A.11.5.1	AC-7
E4 - Modify authentication mechanisms	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E5 – Create user account	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E6 - Modify user account	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E7 - Create role	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E8 - Modify role	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E9 - Grant/revoke user privileges	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2

Table 1 – Foundation Events for Logging and Security Framework					
Security Events and Actions	**PCI DSS 10.2**	**SOX (COBIT)**	**HIPAA (NIST 800-66)**	**IT Security (ISO 27001)**	**FISMA (NIST 800-53)**
E10 - Grant/revoke role privileges	10.2.5	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E11 - Privileged commands	10.2.2	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2
E12 - Modify audit and logging	10.2.6	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2 AU-9
E13 - Objects: Create object Modify object Delete object	10.2.7	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2 AU-14
E14 - Modify configuration settings	10.2.2	DS5.5 DS5.6 DS9.2	164.312(c)(2)	A 10.10.1	AU-2

FRAMEWORK APPROACH

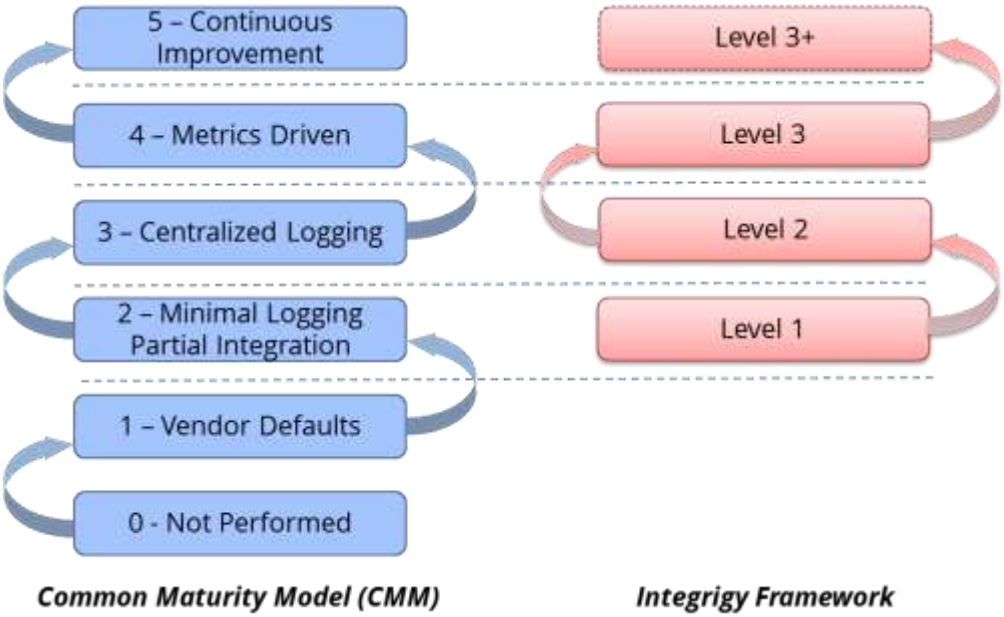
Integrigy's framework has three levels of maturity. Not all organizations will start at the same level. Which level a client starts at depends on the infrastructure and policies an organization already has in place. Integrigy's experience is that using this approach will give both specific guidance as well as vision.

The levels are:

- **Level 1** – Enable basic logging for PeopleSoft system administration and implement a best practices checklist for monitoring and auditing of security-sensitive events. Implementation focus is on DBAs and application administrators and protecting the tables that define security within the PeopleSoft application.
- **Level 2** – Send basic log data to a centralized logging solution outside the Oracle Database and PeopleSoft. Implementation focus is on IT security and internal auditors and their meeting the basic requirements.
- **Level 3** – Send PeopleSoft functional and additional database logs to the centralized logging solution. Implementation focus is on securing sensitive data such as Personally Identifiable Information (PII) to assist IT security and internal auditors to meet advanced requirements for compliance and automation. Securing PII data is commonly done to meet specific requirements for compliance PCI, SOX, HIPAA and ISO 27001.

▪

Figure 1 - Integrigy Framework Compared to Common Maturity Model



Level 1

The first level focuses on logging and basic monitoring and auditing of security-sensitive events. Logging, monitoring, and auditing are separate but related disciplines. Logging provides the data for both monitoring and auditing. In the Framework's first level, the optional logging functionality is enabled. This is functionality not enabled by the default PeopleSoft installation and is commonly not used. Once this functionality is in place, the Framework then presents a best practice checklist for security monitoring and auditing for PeopleSoft. For those customers considering a security monitoring and auditing program, this should be an ideal starting point. A key concern with Level 1 is to secure the tables that define security within the PeopleSoft Application.

Level 2

The second level of maturity focuses on integrating with a centralized logging solution. Given the complexity of PeopleSoft and compliance requirements for protection and non-repudiation of log data, a centralized logging solution is required. Once the solution is in place, Level 2 of the Framework presents where and how to start passing log and audit data from PeopleSoft and Oracle Database.

Level 3

The third level of maturity is continuous (hence Level 3+ in the graphic above). Once the basic log data is being passed to a centralized logging solution and/or Security Incident and Event Management (SIEM) system, the Framework presents additional PeopleSoft configurations that can and should be considered for event correlation. A key focus of Level 3 is on securing sensitive data such as Personally Identifiable Information (PII). As well, Level 3 identifies additional database and application server logs to be captured. Level 3 is continuous, as the possibilities of detecting security incidents using event correlation rules and filters are only limited by the data within PeopleSoft. Securing sensitive data should also be a top priority for Level 3.

Figure 2 - Integrigy Framework Auditing and Logging Framework

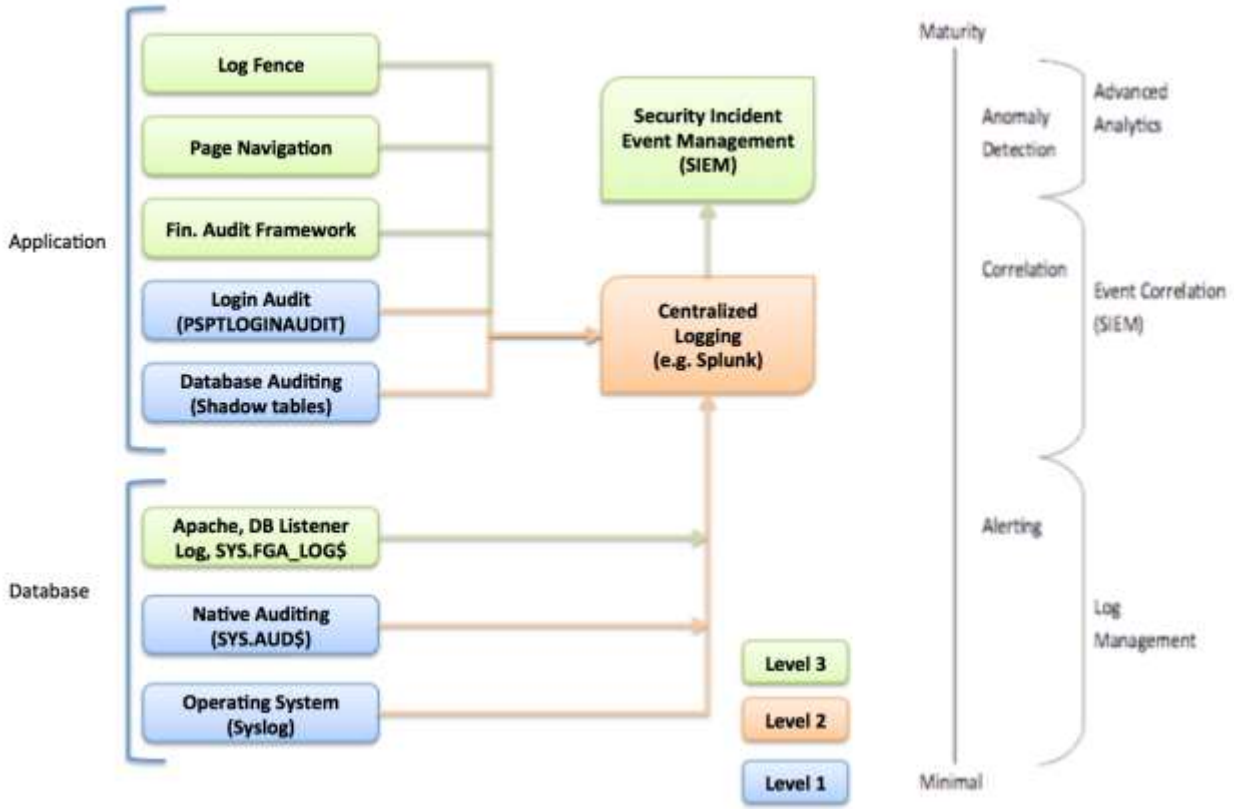
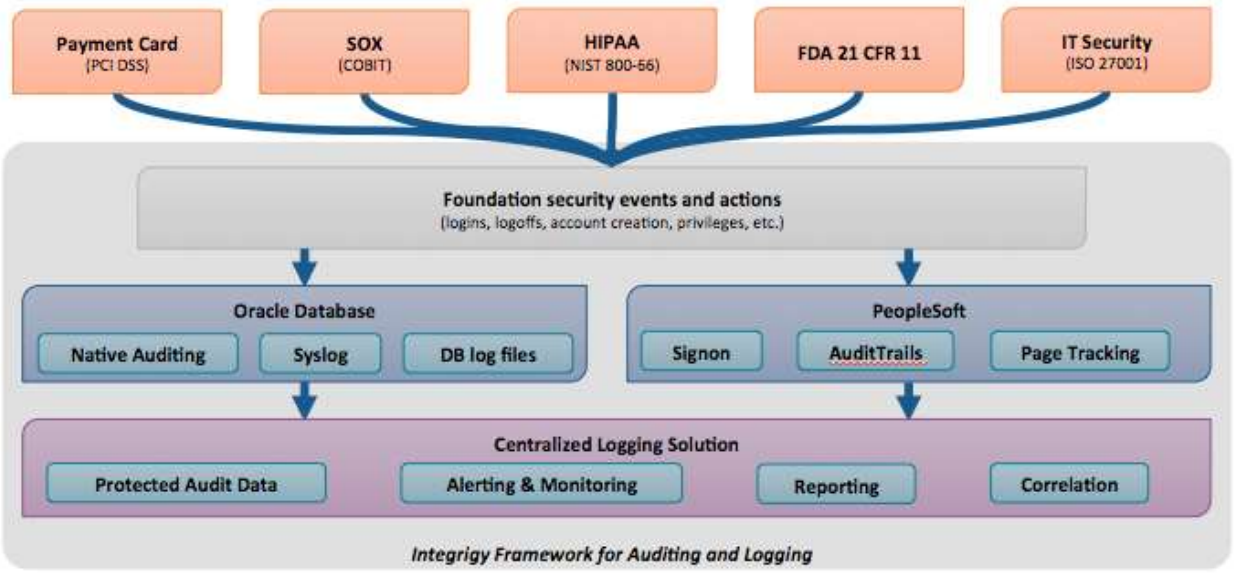


Figure 3 - Integrigy Framework for Auditing and Logging in PeopleSoft



LOG AND AUDIT FUNCTIONALITY

This section reviews the basic log and audit functionality available in PeopleSoft and the Oracle Database. Some of this functionality is enabled by default – some of it is optional and needs to be configured. It should also be noted that more audit and monitoring functionality exists than what is discussed here. The scope of this discussion is limited to what is required to implement Integrigy's framework.

NOTE: This section may be optional if the reader is already familiar with the core auditing and logging functionality in PeopleSoft. The purpose is to provide an overview of the key auditing and logging features used to implement Integrigy's framework.

WHAT IS A LOG?

A "log" is a collection of messages that "paints a picture" of an event or occurrence. The following are general categories of log messages, all of which are important to Integrigy's framework:

- **Informational –** benign event occurrence, for example, a system reboot
- **Debug –** information to aid developers and administrators
- **Warning –** events affecting systems and applications
- **Error –** application or system fault
- **Alert –** something interesting has occurred

A log message has three parts:

1. **Timestamp** – when did the event occur
2. **Source** – server, application or person
3. **Data** – system message, SQL statement, debug code, etc.

OPERATING SYSTEM LOGGING

Most, if not all, PeopleSoft implementations running on UNIX or Linux will have Syslog enabled by the system administrators and/or hosting provider. Syslog is a standard for UNIX and Linux message logging and supports a wide variety of devices, from printers and network routers to database servers. Syslog messages generated by applications or services are sent to a message store on the system or can be delivered to a centralized server built for the specific purpose of log storage and analysis.

The following basic operating system events are assumed collected and available:

- System startup/shutdown
- Logons and attempted logons – IP address, port, time
- Process history and statics

ORACLE DATABASE

Oracle Databases offer a rich set of logging and auditing functionality. For Integrigy's Framework, standard Oracle Database auditing and the capability to send database audit logs to Syslog will be leveraged.

Standard Oracle Auditing

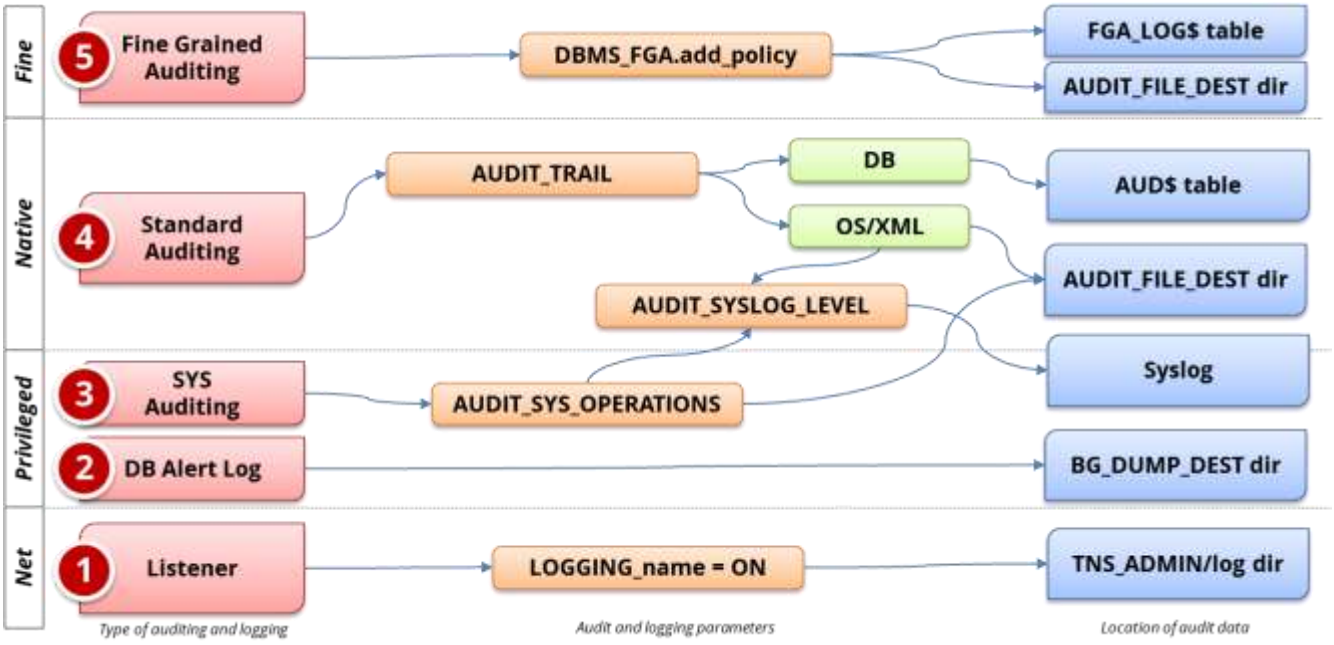
Standard auditing is available in all editions of the Oracle RDBMS. It can be used to audit SQL statements, privileges, schemas, objects and network and multitier activity. Standard auditing must be enabled, and once enabled, a regular program for purging data needs to be implemented.

The variety and volume of data collected by standard auditing can be large, and the output can be directed either to the database itself or files in the operating system outside the database. Either moving logs outside the reach of DBAs, into the operating system or sent to a centralized log server, offers many security benefits. For more information on standard auditing, refer to the reference section of this document.

Database Syslog

As noted earlier, Syslog is a standard for UNIX and Linux logging. Oracle Syslog option is a standard database feature that sends Oracle log data to the native operating system Syslog facility, which in turn can be forwarded directly to a centralized Syslog server or collector. The native Oracle Syslog auditing has minimal performance overhead and provides immediate protection of the audit trail. However, it is possible for the DBA to disable auditing and mitigating controls must be established around possible deactivation of the auditing. For more information on Syslog, refer to the reference section of this document.

Figure 4 - Database Auditing and Logging



PEOPLESOFT

PeopleSoft provides a robust set of default and optional audit functionality. This includes the following:

- Login Auditing (successful/unsuccessful)
- Navigation (Page Level) Auditing – who went where and looked at what?
- Field Auditing – who created or last updated what?
- Database Auditing – who created or last updated what?

For those familiar with PeopleSoft's auditing functionality, this section can be skipped. For those not familiar this section will give an overview. Refer to the PeopleSoft documentation for a detailed review and explanation.

Login Auditing

PeopleSoft login auditing is sent to the table PSPTLOGINAUDIT for both successful and unsuccessful attempts. To enable the Login Audit option use PSADMIN (psSYSADMrv.cfg) to ensure the following domain parameter is set for auditing. On the application server configuration file look in *PS_CFG_HOME*\SYSADMerv*domain_name* for the file PSSYSADMRV.CFG. Locate the parameter 'Enable Login Audit' and Set Enable Login Audit option = Y

Table PSLOGINAUDIT	
Column	**Description**
PT_AUTH_TYPE	0 = Authentication token used 1 = Database authentication 2 = PeopleCode authentication
OPRID	Profile (User Account) in table PSOPRDEFN
PTSIGNONID	User ID used in the attempt. May differ is LDAP is being used.
PT_SIGNON_STATUS	0 = Success 1 = Failure
LASTSIGNONDTTM	Date time of event

Navigation (Page Level) Auditing

PeopleSoft navigation auditing can be enabled. It is not enabled by default. It is enabled by using PeopleTools to add code to the open event of pages such that information is written out to audit logs to log the page name and data records viewed. To enable this functionality refer to the following Oracle support whitepaper: PeopleSoft Security Auditing (Doc ID 1963774.1).

Field Auditing

Field level auditing is delivered by default with PeopleSoft. It is optional in that that functionality is provided but must be enabled on a field-by-field basis. Regardless of what fields are enabled for auditing, all audit records are written to the centralized audit table PSAUDIT.

One important caveat about field level auditing is that only logs activity that occurs within the PeopleSoft user interface. Direct database activity from DBAs and developers using SQL-Plus and/or SQL-Developer will not be detected by field level auditing, nor will field level auditing secure activity occurring within SQR and PeopleSoft's

COBOL programs. Most importantly, however, field level auditing does not support auditing of PeopleTools tables (PT 8.1x, 8.4.x and PT 8.5.x)[1].

Field level auditing logs:

- Who made the change (OPRID)
- Date and Time Stamp of the change
- Type of change: 'A'dd o 'C'hange o 'D'elete
- Record Name (table):
- Field Name in the Record
- The before value
- The after value
- Primary Key of the Record

```
The following SQL identifies fields on records that have field level auditing:

SELECT
F.RECNAME,
F.FIELDNUM,
F.FIELDNAME,
F.USEEDIT,
CASE WHEN BITAND(F.USEEDIT,8) > 0 THEN 'Y' ELSE 'N' END AUDIT_FIELD_ADD, CASE WHEN
BITAND(F.USEEDIT,128) > 0 THEN 'Y' ELSE 'N' END AUDIT_FIELD_CHANGE, CASE WHEN
BITAND(F.USEEDIT,1024) > 0 THEN 'Y' ELSE 'N' END AUDIT_FIELD_DELETE
FROM
SYSADM.PSRECFIELD F
WHERE
F.FIELDNAME = (
SELECT
CASE WHEN (
BITAND(USEEDIT,8) > 0 OR BITAND(USEEDIT,128) > 0 OR BITAND(USEEDIT,1024) > 0
) THEN FIELDNAME ELSE '' END AS FIELD_AUDITED FROM SYSADM.PSRECFIELD
WHERE RECNAME = F.RECNAME
AND FIELDNAME = F.FIELDNAME )
ORDER BY F.RECNAME, F.FIELDNUM;


Audit records will be written to PSAUDIT.
-- Records being audited
SELECT
DECODE(TRIM(AUDITRECNAME),NULL,'NOT ENABLED','ENABLED') AUDIT_STATUS,
PSRECDEFN.RECNAME ,  NVL(TRIM(PSRECDEFN.SQLTABLENAME),'PS_'||PSRECDEFN.RECNAME)
THETABLE ,
TRIM(AUDITRECNAME) AUDITRECNAME,
CASE WHEN BITAND(RECUSE,1) > 0 THEN 'Y' ELSE 'N' END AUDIT_ADD,
CASE WHEN BITAND(RECUSE,2) > 0 THEN 'Y' ELSE 'N' END AUDIT_CHANGE,
CASE WHEN BITAND(RECUSE,4) > 0 THEN 'Y' ELSE 'N' END AUDIT_DELETE,
CASE WHEN BITAND(RECUSE,8) > 0 THEN 'Y' ELSE 'N' END AUDIT_SELECTIVE,
PSRECDEFN.OBJECTOWNERID,
PSRECDEFN.RECDESCR,
PSRECDEFN.DESCRLONG
FROM SYSADM.PSRECDEFN
WHERE  PSRECDEFN.RECTYPE = 0
ORDER BY 1, 2;
```

[1] Can PeopleTools Tables Such As Security Tables Be Audited? (Doc ID 611582.1)

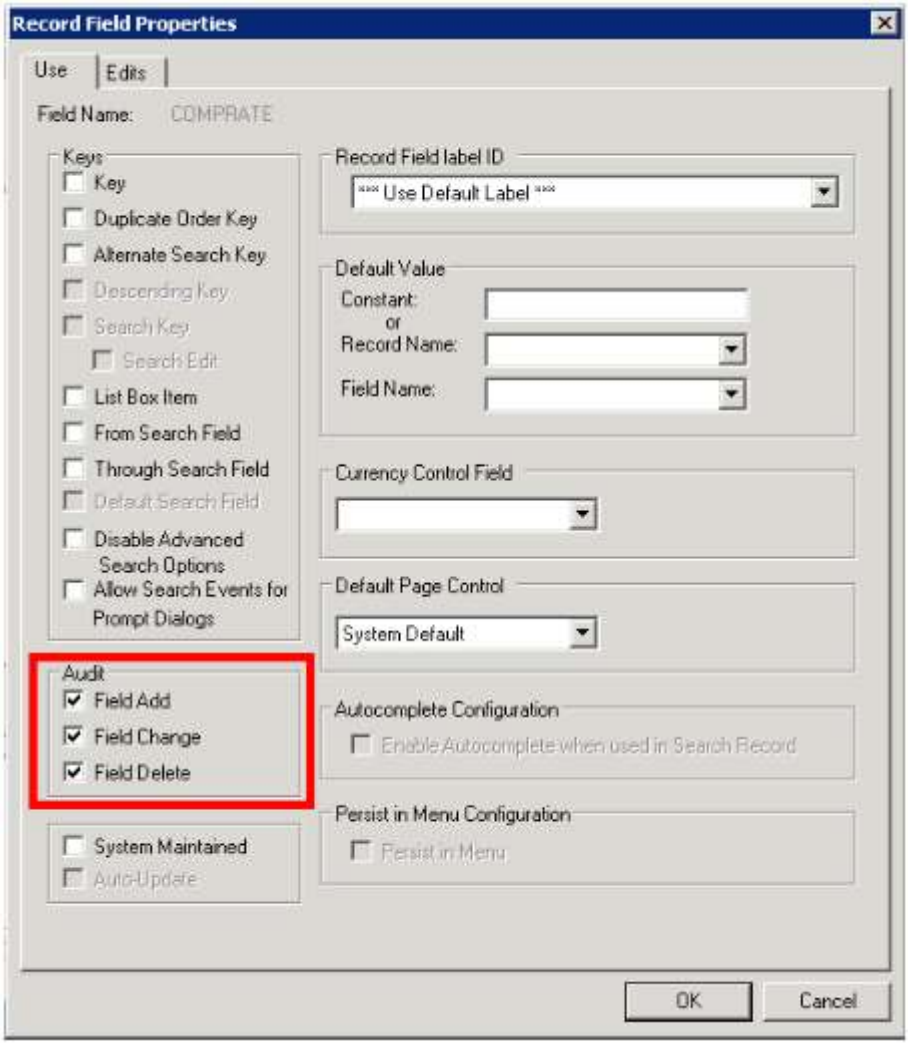


Figure 5 - Example of Field Level Auditing

Database Auditing

Database auditing is an alternative to Field Level Auditing. Database triggers are used to copy information from PeopleSoft base tables into "shadow" audit tables. To enable this functionality, the parameter Enable DB Monitoring must be enabled in PSADMIN. Other steps are also required and are detailed in a forthcoming section of this document.

Whereas PeopleSoft field level auditing logs only activity that occurs within the PeopleSoft user interface, database auditing logs ALL activity regardless if coming from the end-user interface, SQR, COBOL and/or direct database activity from DBAs and developers using SQL-Plus and/or SQL-Developer. Most importantly, however, database auditing will log activity in PeopleTools tables (PT 8.1x, 8.4x and PT 8.5x)[2].

[2] Can PeopleTools Tables Such As Security Tables Be Audited? (Doc ID 611582.1)

A key point with trigger-based auditing is that the database (e.g. Oracle RDBMS) will capture INSERTs, UPDATEs, and DELETEs, not SELECTs. This is the standard functionality of the Oracle RDMBS that triggers are fired only when data is changed, not when it is read. Primarily for this reason, and to reduce overall complexity, the Integrigy's recommendation is to NOT use PeopleSoft Audit triggers and to instead use Oracle Fine Grained Auditing (FGA). FGA allows for targeted auditing for all DML (INSERT, UPDATE, DELETE, SELECT).

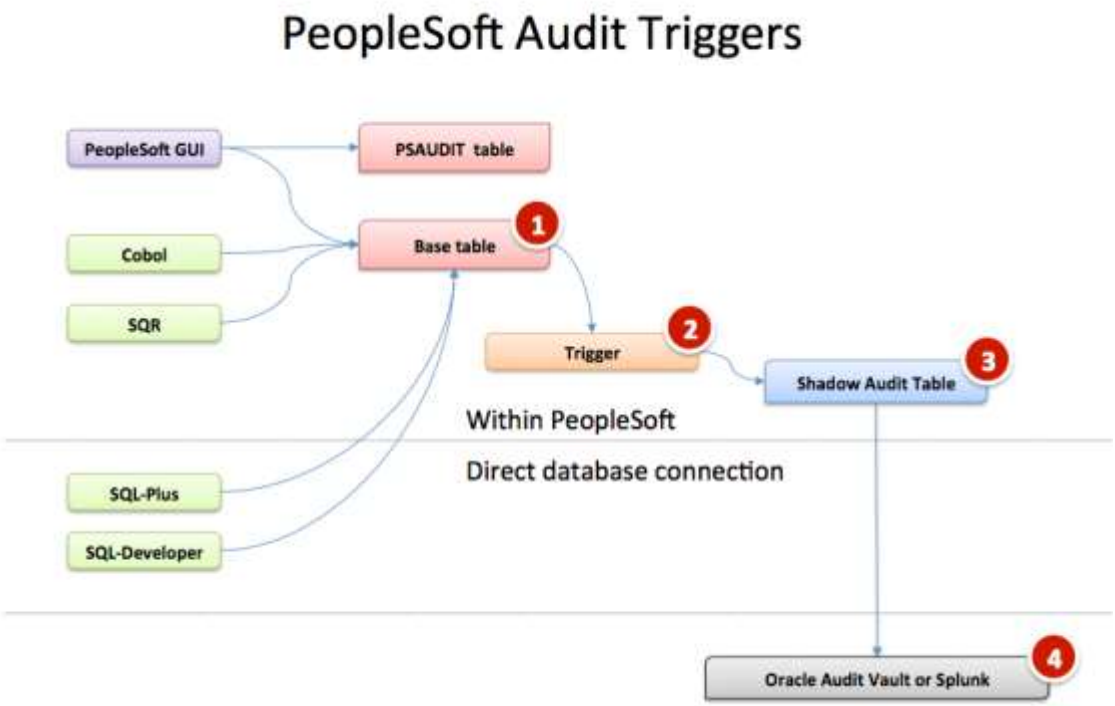


Figure 6 - PeopleSoft Database Auditing

INTEGRIGY FRAMEWORK – LEVEL 1

Level 1 focuses on the basic logging that Integrigy recommends for all PeopleSoft implementations. This logging needs to be in place before proceeding to Levels 2 and 3 of the Framework but assumes a centralized logging solution is not available yet.

The following summarizes the steps to implement Level 1:

1. Oracle Database logging
    a. Enable standard database auditing to the database (AUD$) per Integrigy's recommendations
    b. Enable AUDIT_SYS_OPERATIONS
2. PeopleSoft logging
    a. Enable Signon Auditing
    b. Enable DB Monitoring
    c. Define Fine Grained Audit (FGA) policies for security sensitive tables
3. Define organizational procedures for security monitoring and auditing

DATABASE AUDITING

Database auditing is vital to application logging, and security monitoring as direct database access can be used to circumvent all application controls.

Level 1 assumes there is no centralized logging solution implemented and the database audit data should be written to the database (SYS.AUD$) for monitoring and reporting. Saving audit data to the database is not ideal as the DBA can manipulate the audit data, but provides for much-simplified monitoring and reporting. If a centralized logging solution is implemented, then the database audit data should be written to Syslog per the instructions in Level 2.

As part of the Framework Level 1, we do not recommend enabling extensive auditing of database objects (e.g., tables, indexes, procedures, etc.) creation, modification, or deletion since in a PeopleSoft environment this will generate a significant amount of audit data. The application itself is creating temporary objects, and there are frequent changes due to patching. The SYSADM user is the account used during these activities and mostly originates from the application or database servers; thus the audit trail becomes meaningless.

Steps for Level 1 database auditing:

1. Enable native database auditing and store audit data to the database. In the init.ora file for the instance, set the database initialization parameter **AUDIT_TRAIL** to **a value of 'DB**'. This will write out all logs to the SYS.AUD$ table except for SYS Operations, which are always written to the operating system audit trail.
2. As the SYS user, configure database auditing per *Table 2 – Recommended PeopleSoft Database Auditing*.
3. The SYS.AUD$ table needs to be purged on a periodic basis per your organization's policy requirement. All rows should be backed up before being purged. Purging is configured through the use **DBMS_AUDIT_MGMT**.

4. In the init.ora file for the database instance, enable auditing of the SYS user by setting the database initialization parameter **AUDIT_SYS_OPERATIONS** to **TRUE**. Logs are written to the operating system's native audit trail.

Table 2 – Recommended PeopleSoft Database Auditing				
Framework Event	**Database Object**	**Oracle Audit Statement (audit {};)**	**Resulting Audited SQL Statements**	**Notes**
E1, E2, E3	Session	session	Database logons and failed logons	▪ All database logons and failed logons ▪ This is highly dependent on database usage and application. With application connection pooling, the number of database session is minimized. However, some frequent interface programs may result in large numbers of sessions.
E5, E6	Users	user	create user alter user drop user	▪ All changes to users ▪ Includes all password changes by users - actual password is not captured
E7, E8	Roles	role	create role alter role drop role	▪ All changes to roles ▪ SET ROLE is excluded which is frequently used and would be included if AUDIT ROLE was used
E13	Database Links Public Database Links	database link public database link	create database link drop database link create public database link drop public database link	▪ Creation and deletion of database links
E11, E14	System	alter system	alter system	▪ Changes to the database configuration ▪ Audits killing of sessions, open/closing wallet, and setting of initialization parameters
	Database	alter database	alter database	▪ Change to database and instance state
E9, E10	Grants (system privileges and roles)	system grant	grant revoke	▪ Captures only grants to system privileges and roles ▪ Grants/revokes on database objects will be captured as part of the object creation

Table 2 – Recommended PeopleSoft Database Auditing				
Framework Event	**Database Object**	**Oracle Audit Statement (audit {};)**	**Resulting Audited SQL Statements**	**Notes**
E4	Profiles	profile	create profile alter profile drop profile	▪ All changes to password and resource profiles ▪ Assigning profiles to users will be captured as part of ALTER USER
E9, E10	Directories	grant directory	grant directory revoke directory	▪ Granting of directories
E9, E10	Procedures Packages Functions Libraries Java Objects	grant procedure	grant <procedural type> revoke <procedural type>	▪ Granting and revoking of procedural objects
E9, E10	Object Grants	grant sequence grant table grant type	grant sequence grant table/view grant type revoke sequence revoke table/view revoke type	▪ Granting on sequence, tables, types, and views ▪ Grant table will also audit grant view
E12	Auditing	system audit	audit noaudit	▪ Changes to database auditing
E11, E14	SYSDBA and SYSOPER	sysdba sysoper	All SQL executed with sysdba and sysoper privileges	▪ Actions taken by DBAs – mostly occurs during weekly maintenance window

PEOPLESOFT AUDITING

PeopleSoft's default auditing sends record and field level data to centralized tables (PSAUDIT). Field level auditing has several disadvantages. First, field level auditing secures only activity that occurs within the PeopleSoft user interface. Direct database activity from DBAs and developers using SQL-Plus and/or SQL-Developer will not be detected by field level auditing, nor will field level auditing secure activity occurring within SQR and PeopleSoft's COBOL programs. Most importantly, however, field level auditing does not support auditing of PeopleTools tables (PT 8.1x, 8.4x and PT 8.5x). Consequently, Integrigy's Log and Audit Framework for PeopleSoft uses a database-auditing alternative.

Level 1 of Integrigy's Log and Audit Framework is designed to log and audit the low volume, high-security impact fields that define security rules and relationships and/or can materially affect the confidentiality, integrity, and availability. Level 3 of the Framework is designed to log and audit tables with sensitive information such as Personally Identifiable Information (PII). Integrigy recommends using Oracle Fine Grained Auditing (FGA) for both Level 1 and Level 3. Clients can choose to implement both sets of recommendations at the same time; Integrigy separates Level 1 and Level 3 recommendations only for documentation purposes.

PeopleSoft does offer a database agnostic auditing solution that utilizes triggers and shadow tables. This option is documented in Appendix B and involves defining new shadow audit records in the Application Designer and then generating and deploying database triggers to populate the shadow audit tables. This option is not recommended primarily due to its overall limited efficiency due to its performance overhead and complexity to implement. For most clients, Fine Grained Auditing (FGA), free with the enterprise edition of the Oracle RDBMS, will more than meet their requirements for performance, compliance, and security.

Recommendation: Do Not Use PeopleSoft Database Trigger and Shadow Table Auditing

One difference between FGA and the shadow table auditing is that while FGA can capture the full SQL as well as bind variables, FGA auditing, however, does not maintain the full history of all prior states of a record. For clients needing a complete change history of records, for example, each prior salary or spelling of employee's names, the trigger based shadow table auditing will be to be considered. For clients not using the Oracle RDBMS to support their PeopleSoft implementation, both DB2 and MS SQL-Server offer their variations of FGA.

The recommended steps to configure PeopleSoft auditing for Level 1 of the Framework are as follows:

1. Enable Login Auditing
2. Enable DB monitoring
3. Define FGA policies

Enable Login Auditing

This step is not part of database auditing but is good to do before enabling database auditing. Enabling Login auditing enhances the information capture PeopleSoft logon/off and attempts to the table PSPTLOGINAUDIT.

To enable the Login Audit option, use PSADMIN (psSYSADMrv.cfg) or on the application server configuration file look in *PS_CFG_HOME*\SYSADM*domain_name* for the file psappsrv.cfg. Locate the parameter 'Enable Login Audit' and Set Enable Login Audit option = Y.

Enable DB Monitoring

Enabling DB Monitoring allows the PeopleSoft end-user User Id (PSOPRID) to be automatically passed into the FGA logs. To enable this use PSADMIN (psappsrv.cfg) and set EnableDBMonitoring = 1.

Verification:

1. On the application server configuration file look in *PS_CFG_HOME*\SYSADM*domain_name* for the file psappsrv.cfg (for example in /home/psadm2/psft/pt/8.54/appserv/APPDOM/psappsrv.cfg)
2. Locate the parameter EnableDBMonitoring
3. Make user EnableDBMonitoring = 1

PeopleSoft FGA Auditing

FGA policies can be configured before or after Enable DB Monitoring is enabled. The requirement of enabling DB Monitoring is to allow the PeopleSoft end-user User Id (PSOPRID) to be automatically passed into the FGA log.

FGA policies are created on a table/column basis can and are triggered either by all activity or by only when specific criteria are met – such as when such activity is occurring outside the PeopleSoft application. Most importantly, FGA policies can exclude the PeopleSoft application to exclusively focus on logging database activity by DBAs, developers or anyone else logging directly into the database.

Integrigy's Level 1 recommendations for FGA policies on key security tables can be easily expanded also to include Integrigy's Level 3 recommendations for tables containing sensitive data. It is your choice to implement both sets of recommendations at the same or not.

FGA logs are written to SYS.FGA_LOG$ and can be defined for any combination of DML (INSERT, UPDATE, DELETE, SELECT). The system view DBA_FGA_AUDIT_TRAIL is provided by Oracle for enhanced reporting of SYS.FGA_LOG$.

To meet the requirements of Level 2, SYS.FGA_LOG$ needs to be relayed to a centralized logging solution. Oracle Audit Vault will natively consume FGA audit logs as will other industry standard tools such as Splunk.

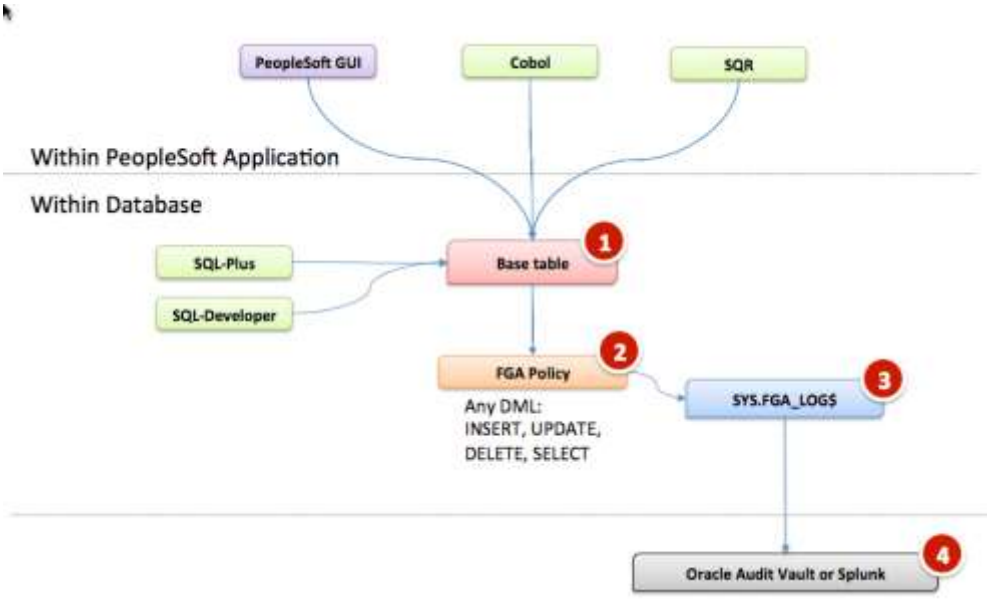


Figure 7 - FGA option for PeopleSoft

More detailed information on PeopleSoft's support of FGA can be found in the following links:

- PeopleSoft FGA support: https://tinyurl.com/n7g67rp
- Oracle RDBMS documentation on FGA: https://tinyurl.com/mu2mh2g

Recommended Technical Approach

FGA can be used to log direct database and/or PeopleSoft activity. The overall recommended technical approach for using FGA with PeopleSoft is to be able to separate direct database logon activity on activity on key tables from native PeopleSoft application activity. For this, two steps are required -

1. Create a database Logon trigger
2. Create FGA policies on target table/columns
3. Define purge processes

Create Logon Trigger

The first step is to create a database logon trigger to set a context variable. This is a standard recommended best practice for using FGA and allows database sessions to be identified (flagged) as either coming from the PeopleSoft application itself (e.g. GUI forms, SQR, Cobol) or developers and DBAs directly accessing the database from their laptops.

Create FGA Policies

The second step is then to create FGA policies against the target tables/columns. These policies can be written only to be fired when the context variable identifies the database session as NOT coming from the PeopleSoft application. The policies can just as easily be written to fire for any user/session activity as well as even targeting specific PeopleSoft users (PSOPRID). Starting with PeopleTools 8.50, the PSOPRID is set in the Oracle database CLIENT_IDENTIFIER attribute, and this attribute can be used in defining FGA policies.

The graphic below summarizes the overall recommended technical approach for using FGA with PeopleSoft for Level 1 of the Framework, with a focus on securing data from direct database logins. Keep in mind that during Oracle's create session (login), Logon triggers fire AFTER the traditional native auditing.

There are performance ramifications of using auditing, but a small number of audits on key tables have proven to be very safe. However, testing should also be done before production.

Define Purge Processes

Once defined and enabled, FGA policies are written out to SYS.FGA_LOG$. This table may with appropriate privileges be manually purged. However, it is highly recommended to use the DBMS_AUDIT_MGMT package to setup a rolling purge per your organization’s policies and requirements. It is also recommended to monitor the size of the SYS.FGA_LOG$ table as well.

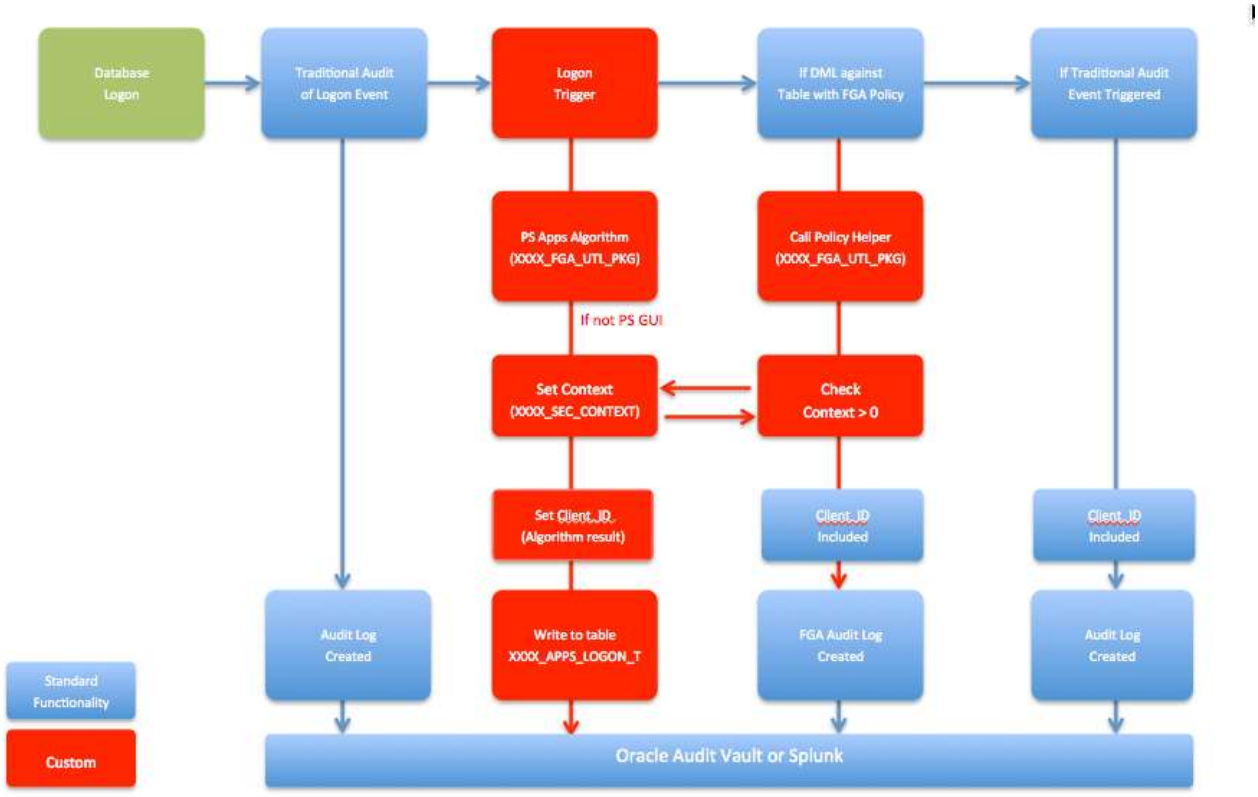


Figure 8 - Recommended Technical Approach

Recommended FGA Policies

Level 1 of the Framework focuses on the security sensitive tables. These are the tables the either define security rules and policies within the PeopleSoft application and/or are the source key security activities (e.g. login/logoff/unsuccessful logins).

The following table maps key PeopleSoft security sensitive tables to the Framework. Not all tables need to be included for every client.

PeopleSoft Table Mappings to Framework	
Integrigy Framework	**Table**
E1 - Login	PSOPRDEFN, PSPTLOGINAUDIT, PSACCESSLOG
E2 - Logoff	PSACCESSLOG
E3 - Unsuccessful login	PSPTLOGINAUDIT, PSACCESSLOG
E4 - Modify authentication mechanisms	PSSECOPTIONS
E5 - Create user account E6 - Modify user account	PSOPRDEFN
E7 - Create role E8 - Modify role	Permission lists and roles: PSAUTHITEM, PSROLECLASS, PSROLEDEFN
E9 - Grant/revoke user privileges	PSROLEUSER, PSOPRCLS
E10 - Grant/revoke role privileges	PSPGEACCESSDESC

PeopleSoft Table Mappings to Framework	
Integrigy Framework	**Table**
E12 - Modify audit and logging	PeopleTools: PSRECDEFN, PSRECFIELD, Financials: AUDIT_CNTRL_TBL Audit records: PSAUDIT and all shadow tables
E13 - Objects: Create object Modify object Delete object	PSRECDEFN, PSRECFIELDALL, PSPNLDEFN, PSMENUITEM, PSPROJECTITEM
E14 - Modify configuration settings	PSOPTIONS , PSSECOPTIONS, PS_SJT_PERSON, PSSECNODEOPR, PSSEC_PPC_OPTN, PSIBPROFILE, PS_HCR_SCRTY_INSTL, PS_INSTALLATION_FS, PS_INSTALLATION_BN, PS_GPSINSTALLATION, PS_EP_INSTALLATION, PS_GP_INSTALLATION, PS_US_INSTALLATION, PS_INSTALLATION, PS_INSTALLATION_HR, PS_JPMINSTALLATION, PS_INSTALLATION_PA, PS_INSTALLATION_PY, PS_HRSINSTALLATION, PS_TL_INSTALLATION, PS_INSTALLATION_PB, PS_INSTALLATION_AA, PS_INSTALLATION_AD, PS_INSTALLATION_AV, S_INSTALLATION_SA, PS_INSTALLATION_CC, PS_INSTALLATION_FA, PS_INSTALLATION_SR, PS_BUS_UNIT_OPT_HR, PS_SCRTY_SET_TBL , PS_SCRTY_QUERY, PS_SCRTY_TBL_DEPT, w_SCRTY_DEPT, PSPTSCRTY_ADS_A, PSRF_RSCRTY_TBL, PS_SCRTY_ACC_GRP, PSPTSCRTY_ADS_P, PS_SCRTY_SRCHGRP, PS_SCRTY_TBL_INST

Level 1 of the Framework assumes that all updates and/or deletes from direct database activity outside PeopleSoft application will be logged. Optionally, clients could choose to log ALL activity occurring both inside PeopleSoft and from direct database connections. If All activity is required to be logged, high-volume tables such as login/logout activity should be avoided except for attempts to later their logs through DELETEs or UPDATEs.

It is possible to define FGA policies to include internal PeopleSoft activity and/or specific target specific PeopleSoft users. This would be done by using the referencing the CLIENT_IDENTIFIER in the FGA policy's audit condition (e.g. where CLIENT_IDENTIFIER in SYS.V$SESSION = 'MMILLER'). Additional conditional logic is also possible, using combinations of IP addresses, Usernames and/or time-of-day. Refer to the Oracle documentation on defining FGA policies (see link above) for more information on defining FGA policies.

Key points for Level 1 FGA auditing:

- To start define FGA policies and enable them, you do NOT need to first enable traditional database auditing. For example, you do NOT need to set audit_trail=DB in v$parameter and bounce the database. You can start using FGA at any point in time.

- Target all columns – The focus of Level 1 are changes to key security data as the result of direct database activity. All columns in the recommended tables should be in scope.
- For the most part, focus on changes to data NOT reads (SELECTs). The exception is for high volume tables such as the login audit tables where INSERTS should be audited.
- Include the SQL that that triggered the FGA policy as well as the bind variables. To do this set audit_trial = DBMS_FGA.DB +DBMS_FGA.EXTENDED

Recommended Level 1 FGA Policies					
#	**Frame work**	**DB Table**	**Description**	**Target Columns**	**DML**
1	E14	PS_PRCSDEFN	Process Definition	NULL (default all columns)	INSERT, UPDATE, DELETE
2	E1, E2	PSACCESSLOG*	Login history	NULL (default all columns)	UPDATE, DELETE
3	E14	PSCLASSDEFN	Permissions Lists Definition	NULL (default all columns)	INSERT, UPDATE, DELETE
4	E14	PSMENUDEFN	Menu Definition	NULL (default all columns)	INSERT, UPDATE, DELETE
5	E13	PSMENUITEM	Menu Item	NULL (default all columns)	INSERT, UPDATE, DELETE
6	E4, E5	PSOPRDEFN	User definition	NULL (default all columns)	INSERT, UPDATE, DELETE
7	E1, E3	PSPTLOGINAUDIT*	Login history	NULL (default all columns)	UPDATE, DELETE
8	E12, E13	PSRECDEFN	Record Definition	NULL (default all columns)	INSERT, UPDATE, DELETE
9	E7, E8	PSROLECLASS	Role Classes	NULL (default all columns)	INSERT, UPDATE, DELETE
10	E7, E8	PSROLEDEFN	Role Definition	NULL (default all columns)	INSERT, UPDATE, DELETE
11	E9	PSROLEUSER	Role User	NULL (default all columns)	INSERT, UPDATE, DELETE
12	E4, E5, E6, E14	PSSECOPTIONS	Password controls	NULL (default all columns)	UPDATE, DELETE
13	E14	PSWEBPROFILE	Web Profile	NULL (default all columns)	INSERT, UPDATE, DELETE

* Recommended only for logging direct database activities.

Use the example below to create FGA policies for the recommended tables above. Additional tables are listed in Appendix A. The example below creates an FGA policy on the table `PSSECOPTIONS` that defines the password controls for PeopleSoft. Note that the audit_condition parameter has been set to NULL to log any change to these parameters regardless of being done through the application or from a direct database session. Depending on your requirements you can either leave audit_condition = NULL or specify a condition.

To define FGA policy conditions, Integrigy recommends keeping FGA policy definitions clean and simple. Unless a condition is very simple (e.g. USERNAME != 'SYSADM') it is recommended to use a utility PL/SQL package to define functions to call. For example, such a utility package could hold functions to determine if a session is coming from a direct database login or from the PeopleSoft application itself.

```
DBMS_FGA.ADD_POLICY (object_schema      =>  'SYSADM',
       object_name         =>  'PSSECOPTIONS',
       policy_name         => 'XXXX_FGA_NOT_GUI_PWD_OPTS',
       audit_condition     =>  NULL,
       audit_column        =>  NULL,
       handler_schema      =>  NULL,
       handler_module      =>  NULL,
       enable              =>  TRUE,
       statement_types     =>  'INSERT, DELETE, UPDATE',
       audit_trail         =>  DBMS_FGA.DB +DBMS_FGA.EXTENDED ,
       audit_column_opts   =>  DBMS_FGA.ANY_COLUMNS);
```

Useful SQL

-- Defined FGA policies (you may need privileges granted to use)

```
SELECT POLICY_NAME, ENABLED, OBJECT_SCHEMA, OBJECT_NAME,POLICY_COLUMN,
POLICY_TEXT
FROM SYS.DBA_AUDIT_POLICIES
ORDER BY 1,2;
```

-- FGA Log (you may need privileges granted to use)

```
SELECT * FROM DBA_FGA_AUDIT_TRAIL;
```

Recommended Monitoring and Alerts

For Level 1, the assumption is that centralized logging and analysis tools and/or a SIEM are not available. Recommendations are made below for what to monitor. Who to notify in case of a monitoring alert is not possible to recommend because it will be unique to each client site and implementation.

The sources of what needs to be monitored specifically include the Oracle RDBMS audit sources SYS.AUD$ for traditional database auditing and SYS.FGA_LOG$ for FGA auditing. Oracle combines both these source into a view DBA_COMMON_AUDIT_TRAIL. Each of the PeopleSoft shadow audit tables would also need to be monitored. How these sources are monitored, depend on tools available.

It is assumed that clients have reporting tools capable of creating custom and scheduling reports for email delivery – possibly even using the PeopleSoft Scheduler. Whether the alerts are sent immediately or in the form of a daily summary should be determined by each customer's unique risk profile.

Our recommended security monitoring and auditing alerts (Table 4) are by no means conclusive. Simple things can trigger serious high-risk security events and can differ between PeopleSoft implementations. As such, the table below should be seen as much as a starting point as it is an educational tool. What to monitor for and whom to notify will largely be determined by each client's unique risk profile.

Table 4 – Level 1 Security Monitoring and Auditing Alerts

Frame work	What to Monitor For	Description	Source
E1	Direct database logins (successful or unsuccessful) to key database accounts	Direct database attempts, attempts to connect other than through PeopleSoft, should all be investigated - especially for the SYS, SYSTEM, SYSADM and PEOPLE.	SYS.AUD$
E1, E11	User ADMINISTRATOR or User with Power User Roles successful logins	Each login of the ADMINISTRATOR or Power User Roles (see table below) should be logged and reviewed. Daily support should not be done through this account.	PSPTLOGINAUDIT
E1, E11	Generic seeded application account logins	Except for the GUEST accounts, immediate action should be taken if there is attempted login to one of the accounts listed Table below of seeded generic users.	PSPTLOGINAUDIT
E1, E11	Unlocking of generic seeded application accounts	The accounts listed in Table 5 "Default PeopleSoft Users" should always be end-dated. If the end-date for one of these accounts changes, immediate action is required.	PSOPRDEFN shadow audit table

Table 4 – Level 1 Security Monitoring and Auditing Alerts			
Frame work	**What to Monitor For**	**Description**	**Source**
E1 E2	Login/Logoff	Basic login/logoff of user from PeopleSoft	PSPTLOGINAUDIT
E3	User ADMINISTRATOR or User with Power User Roles - unsuccessful login attempts	Multiple unsuccessful login attempts for ADMINISTRATOR or a user with a Power User Role should be considered as a security event. These attempts can also lock the SYSADMIN user. Locking this user can cause applications issues.	PSPTLOGINAUDIT
E4	Modify authentication configurations to database	Database profiles enforce password practices. Changes to how passwords are created, used and validated need to be audited.	Database Profile statements in SYS.AUD$
E4	Modify authentication configurations to PeopleSoft	How PeopleSoft authentication occurs (local or SSO) and if local, how passwords are controlled all need to be logged and audited.	Changes to the audit tables for: PSSECOPTIONS, PSSECNODEOPR, PSWEBPROFILE
E6	New database accounts created	Any changes to the standard PeopleSoft database accounts or creation of new accounts should be audited. Such changes are rare and can indicate inappropriate activity.	SYS.AUD$
E9, E10, E12, E13, E14	Updates audit tables other than by the trigger	The tables recommended to be configured for Audit Trail should not change on a regular basis. Any change to these tables should be alerted or reported per client's risk polices.	FGA_LOG assuming FGA polices are being used.
E12	Turning off or disabling the audit triggers	Disable defined trigger for auditing	SYS.AUD$ for disabling of the triggers defined in in PSTRIGGERDEFN. Assumed are auditing with Audit Alter Trigger.
E12	Turning audit sys operations off	If enabled, disabling audit sys operations is a security event.	V$PARAMETER for "audit_sys_operations"
E12	Turning native database audit off	Disabling database auditing off is a security event.	V$PARAMETER for auditing
E12	Disable/Enable and Alerting of FGA Policies	Disabling or altering FGA policies is a security event	AUDIT EXECUTE on DBMS_FGA BY ACCESS

Table 4 – Level 1 Security Monitoring and Auditing Alerts			
Frame work	**What to Monitor For**	**Description**	**Source**
E1, E2, E3	Log on/off successful/unsuccessful	If Oracle Data Vault is used, track ingress and egress activity.	Oracle Data Vault Logs

Seeded Generic Accounts		
BELHR	JCADMIN1	PSJPN
CAN	NLDHR	PSPOR
CFR	PS	TIME
CNHR	PSCFR	UKHR
ESP	PSDUT	UKNI
FRA	PSESP	USA
FRHR	PSFRA	HSHR
GER	PSGER	WEBGUEST
GRHR	PSINE	WEBMODEL

PeopleSoft Power User Roles		
ADMINISTER_SECURITY	MAINTAIN_SECURITY	PTPP_PORTAL_ADMIN
APPLICATION_DESIGNER	MANAGE_INTEGRATION_PROCESS	QUERY
APPLICATION_ENGINE	MANAGE_INTEGRATION_RULES	QUERY_MANAGER
CUBE MANAGER	MASS_CHANGE	TI_INTEGRATION
DATA_MOVER	NVISION	TREEMANAGER
DEFINITION_SECURITY	OBJECT_SECURITY	UTILITIES
FPY_INTEGRATION	PORTAL_ADMIN	WEB_PROFILE
FT_INTEGRATION	PROCESS_SCHEDULER	WORKFLOW_ADMINISTRATOR
IMPORT_MANAGER)	ADMINISTRATOR	

CONSIDER ORACLE DATABASE VAULT (OPTIONAL)

The PeopleSoft database should not be directly accessed by personnel using the SYSADM user except for specific support tasks. Personnel attempting to use SYSADM schema outside of the PeopleSoft application to alter security sensitive tables should trigger an immediate and real-time security alert. While auditing can assist with this, ideally, the Oracle Database Vault should be considered to both prevent such activity as well as to helping to log attempts for rogue access and updates. Database Vault is a separately license tool, and PeopleSoft is fully certified for use with Oracle Database Vault. Use of Database Vault does not reduce the need for auditing, and it is a complementary tool to assist with Defense-in-Depth.

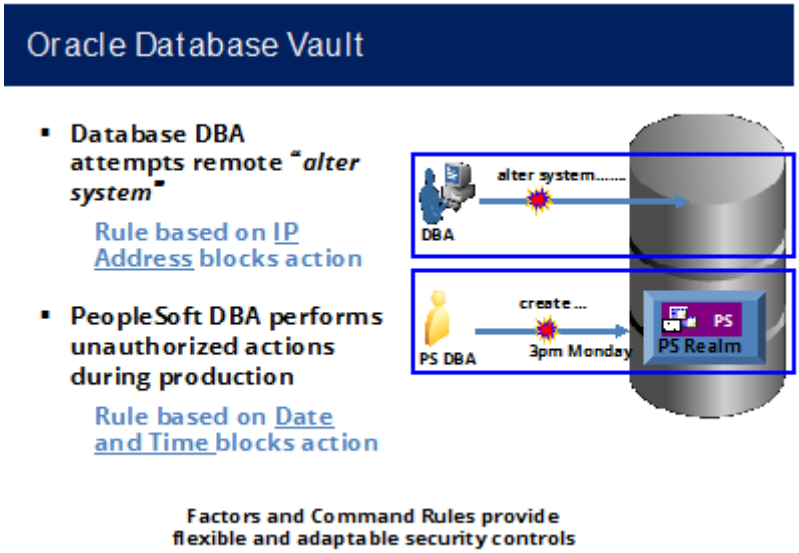


Figure 9 - Data Vault with PeopleSoft

For a detailed explanation of Oracle Database Vault refer to the PeopleSoft documentation here: https://tinyurl.com/k2lh7hr

INTEGRIGY FRAMEWORK – LEVEL 2

The second level of the Framework focuses on integrating with and/or building a centralized logging solution if such a solution does not exist. Such solutions are commonly built using enterprise logging solutions such as Splunk, HP ArcSight, RSA enVision, or Q1 Radar. There are a number of commercial and open-source solutions that can support all the logging and auditing in the Integrigy Framework. For Integrigy's framework, the specific tool is used is not important. What is important is the solution provides (1) ability to accept logs from Syslog, database connections, and reading files, (2) security and archiving of log data, and (3) unified alerting and reporting capabilities.

Centralized logging solutions protect the log data. Non-repudiation and division of duties are achieved by removing log data from each source and storing it in a secure, central location. Consolidating an organization's log data also offers significantly more options for creating security alerts that cross application, team, and geographic boundaries. Centralized logging is also a key requirement for security standards including PCI and HIPAA.

Once the foundation of centralized logging is created with Level 2, an organization can proceed to Level 3. Contact Integrigy with questions and/or assistance with specific centralized logging tools and/or vendors.

Level 2 Tasks

1. Implement centralized logging solution if it does not exist
2. Redirect database logs to centralized logging
3. Configure database connector and send PeopleSoft Sign-on and shadow audit activity to centralized log collector.
4. Transition Level 1 alerts and build additional Level 2 alerts
5. Expand recommended alerts from Appendix A

IMPLEMENT CENTRALIZED LOGGING SOLUTION

The installation and configuration of tools such as Splunk (Free or Enterprise) or HP ArcSight is beyond the scope of this paper. The first requirement for Level 2 is for such a solution to be in place.

REDIRECT DATABASE LOGS TO CENTRALIZED LOGGING

Writing logs to the operating system is more secure for many reasons, including providing a separation of duties between DBAs and system administrators. There are two steps:

1. To route, Oracle database audit logs to the operating system instead of the database set **AUDIT_TRAIL** parameter to **OS** and set **AUDIT_FILE_DEST** to provide a location to write the log files**.**
2. Write logs using the Syslog format. In the init.ora file for the instance, set the **AUDIT_TRAIL** parameter to **OS** and **AUDIT_SYSLOG_LEVEL** to 'LOCAL1.WARNING' or another valid Syslog setting. This setting may be used by the logging server to classify the event.

Transition Level 1 Alerts and Build Additional Level 2 Alerts

As much as possible transition all alerting built for Level 1 to the centralized logging solution. Alerting out of the logging solution (or SIEM) will be more efficient and can provide event correlation capabilities. Moreover, as more alerts will be built, it will consolidate alerting into a single tool.

As with Level 1, the table below is by no means conclusive. Simple things can trigger serious high-risk security events. As such, the table below should be seen as much as a starting point as it is an educational tool. What to monitor for and whom to notify will largely be determined by each client's unique risk profile.

Table 7 – Level 2 Security Monitoring and Auditing Alerts

Event	What to Monitor For	Description	Source
E1	Successful or unsuccessful login attempts to PeopleSoft without network or system login	Logins or attempts to login into PeopleSoft without first logging onto the network or gaining access to the building should be flagged and investigated.	PSPTLOGINAUDIT
E1	Successful or unsuccessful logins of named database user without network or system login	Named database accounts, those associated with staff and employees for the purposes of support should be monitored for if the user has first logged on to the network and/or gained access to the building.	Database log
E3	Horizontal unsuccessful application attempts – more than five users more than five times within the hour	Attempts to brute force groups of users should be alerted and investigated. This alert may be based per IP address or other system identifier. The specific alert threshold will be unique to each client.	PSPTLOGINAUDIT
E3	Horizontal unsuccessful direct database attempts – more than five users more than 5 times within the hour	Attempts to brute force groups of users should be alerted and investigated. This alert may be based per IP address or other system identifier. The specific alert threshold will be unique to each client.	Database log
E9	End-users granted System Administration Roles	End-users gaining access to the highly privileged Power User Roles (See table below) should be carefully reviewed.	Audit tables for: PSROLEUSER, PSOPRCLS
N/A	Monitor for database attacks	The following standard Oracle error messages may indicate a potential database attack: ORA-29532, ORA-28000, ORA-24247,	Database log

Table 7 – Level 2 Security Monitoring and Auditing Alerts			
Event	**What to Monitor For**	**Description**	**Source**
		ORA-29257, ORA-01031	

PeopleSoft Power User Roles		
ADMINISTER_SECURITY	MAINTAIN_SECURITY	PTPP_PORTAL_ADMIN
APPLICATION_DESIGNER	MANAGE_INTEGRATION_PROCESS	QUERY
APPLICATION_ENGINE	MANAGE_INTEGRATION_RULES	QUERY_MANAGER
CUBE MANAGER	MASS_CHANGE	TI_INTEGRATION
DATA_MOVER	NVISION	TREEMANAGER
DEFINITION_SECURITY	OBJECT_SECURITY	UTILITIES
FPY_INTEGRATION	PORTAL_ADMIN	WEB_PROFILE
FT_INTEGRATION	PROCESS_SCHEDULER	WORKFLOW_ADMINISTRATOR
IMPORT_MANAGER)	ADMINISTRATOR	

INTEGRIGY FRAMEWORK – LEVEL 3

Level 3 builds on the connectivity and basic centralized logging established in Level 2. The objective of centralized logging is to consolidate logs from all applications and technologies. While PeopleSoft is but one application, as an Enterprise Resource Planning (ERP) application, it is the cornerstone of most business processes. This is why the objective of Level 3 is the integration of PeopleSoft functional logs with the centralized logging solution as well as having a heavy focus on the of sensitive data by logging and auditing the accessing of sensitive data outside the PeopleSoft application through direct database connections.

Level 3 is also a continuous improvement loop. Once a baseline is established from which alerts and reports are used to report anomalies, as business processes change, tolerances and alerts need to be adjusted to the new baseline. As well, the possibilities of new security alerts and audits are limited by the data consolidated into the centralized logging solution from PeopleSoft, ticket systems, password vaults, network, badging systems, or any other sources capable of producing logs.

Throughout this document, the recommended logging alerts are all able to be mapped back to PCI, HIPAA, NIST 800-53, IS0 27000, and SOX (COBIT). Once Level 3 is reached, efforts should be spent to automate compliance tasks.

ADDITIONAL DATABASE AND APPLICATION LOGS

Each log management or SIEM vendor will have their set of log parsers and capabilities. The recommendation for Level 3 is to send additional database and web server logs to assist with additional logging for who is coming into PeopleSoft, from where and when.

Apache Logs

Apache server logging is defined in the Apache configuration file (HTTPD.CONF). Refer to WebLogic documentation or OHS (Apache) log configuration and setup. Integrigy recommends the default log setting of 'warn'.

Apache Log Levels	Description
emerg	Emergencies, system is not useable
alert	Action must be taken
crit	Critical conditions
error	Error conditions
warn	**Warning conditions - Default**
notice	Normal but significant condition
info	Information
debug	Debug-level messages

Database Listener and Alert Logs

The database listener log provides information regarding database connections, for example, IP addresses of clients, and it should be sent to the centralized logging solution. Within the listener's control file ($TNS_ADMIN/listener.ora), confirm that logging is enabled (LOG_STATUS = On) and the location of the listener log (parameter = LOG_DIRECTORY_listener_name).

Each database has an alert.log. The alert log of a database is a chronological log of messages and errors, including internal errors (ORA-600/7445), corruption errors , and deadlock errors (ORA-60), administrative operations, and SQL*Plus statements STARTUP, SHUTDOWN, ARCHIVE LOG, and RECOVER.

Code	Message
ORA-29532	Java call terminated by uncaught Java exception
ORA-24247	Network access denied by access control list (ACL)
ORA-28000	The account is locked
ORA-29257	Host unknown
ORA-01031	Insufficient privileges

The location of the alert.log can be found using the either of following SQL -

```
SELECT NAME, VALUE FROM V$PARAMETER WHERE NAME = 'diagnostic_dest';
SELECT * FROM V$DIAG_INFO;
```

Starting with Oracle 11gR2, the Oracle database alert log is available in the SYS.X$DBGALERTEXT table. For example -

```
SELECT * FROM SYS.X$DBGALERTEXT WHERE MESSAGE_TEXT LIKE '%ORA-%' GROUP BY
MESSAGE_TEXT;
```

To monitor just critical errors, starting with 11gR2 the view V$DIAG_CRITICAL_ERROR can be used.

```
SELECT * FROM V$DIAG_CRITICAL_ERROR;
```

Use V$DIAG_ALERT_EXT to Monitor both Alert and Listener Log

Also starting with 11gR2 it is possible monitor both the Alert and Listener logs using the system table V$DIAG_ALERT_EXT.

To query just the alert.log:

```
SELECT *FROM V$DIAG_ALERT_EXT
WHERE TRIM(COMPONENT_ID)='rdbms';
```

To query just the Listener.log:

```
SELECT *
FROM V$DIAG_ALERT_EXT
WHERE TRIM(COMPONENT_ID)='tnslsnr';
```

Add Program Name to Audit Trails

The Program Name attribute in V$SESSION (V$SESSION.PROGRAM) is not by default passed to Oracle's audit logs. It can be optionally included. To do so, apply Patch 7023214 on the source database. After the patch is applied, the following event needs to be set:

```
ALTER SYSTEM SET
   EVENT='28058 trace name context forever'
   COMMENT='enable program logging in audit trail' SCOPE=SPFILE;
```

The table below summarizes key session attributes (V$SESSION) that are passed/not passed to Oracle auditing. Starting with PeopleTools 8.50, it is possible to monitor PeopleSoft technology, PeopleCode events and service operations using Module and Action in V$SESSION. To enable this functionality, EnableAEMonitoring must be set to 1 in PSAPPSRV.cfg configuration file. However, for neither of these fields is passed into Fine Grained Auditing

(critical if you are using Oracle Audit Vault). For more information on Monitoring Module and Action refer to the PeopleSoft documentation here: https://tinyurl.com/kobs3z7

Session Attribute (V$SESSION)	Description	Traditional Auditing (SYS.AUD$)	Fine Grained Auditing (SYS.FGA_LOG$)
CLIENT_IDENTIFIER	End user username	CLIENTID	CLIENTID
CLIENT_INFO	Concatenated application log string	Not passed	Not passed
MODULE	Application program, module, application component or service	Not passed	Not passed
ACTION	Business action being executed, page, code event, location within program	Not passed	Not passed
PROGRAM	Operating system program name that established database session	Not passed*	Not passed*

*Can be passed. Refer to Oracle RDBMS Patch 7023214

Use PSOPRID for Level 3 Auditing Rules and Alerts

Before PeopleTools 8.50, the PeopleSoft Operator ID (PSOPRID) was populated in the CLIENT_INFO column in the system view V$SESSION. Starting with PeopleTools 8.50, the PSOPRID is now additionally set in the Oracle database CLIENT_IDENTIFIER attribute. To enable this functionality, EnableAEMonitoring must be set to 1 in PSAPPSRV.cfg configuration file. This means that PeopleTools is automatically populating the PSOPRID (PeopleSoft end-user UserId) into the Oracle RDBMS session attribute 'CLIENT_IDENTIFIER'. For PeopleSoft application sessions, this allows for very powerful correlation and auditing options as the CLIENT_IDENTIFIER is also automatically passed into the native Oracle audit steam (e.g. SYS.AUD$).

The CLIENT_ID is an application context within the database. Application contexts are name-value pairs that the Oracle Database stores in memory. Consider application contexts as global variables that hold information for the duration of a session; they are not persistent.

The CLIENT_ID context is NOT the same as the CLIENT_INFO context. The essential difference between the two is that CLIENT_INFO is set with the DBMS_APPLICATION_INFO package and is only visible in the V$SESSION view. The CLIENT_ID context is set with DBMS_SESSION.SET_IDENTIFIER and is also visible in the V$SESSION view in the column CLIENT_IDENTIFIER, but more importantly, CLIENT_ID is written out to the following Oracle Audit logs:

- DBA_AUDIT_TRAIL (SYS.AUD$)
- DBA_FGA_AUDIT_TRAIL (SYS.FGA_LOG$)
- DBA_COMMON_AUDIT_TRAIL

Financials Audit Framework

The Financials Audit Framework is a separate audit engine unique to PeopleSoft Financials. Adding the Financials Audit Framework should be considered once Level III is reached. Setting up the Financials Audit Framework uses the following components and uses the table AUDIT_CNTRL_TBL to store the configurations:

- Enable Audit Logging (FS_AUDITLOG_ENABLE)
- Search Audit Logs (FS_AUDITLOG_SEARCH)
- Purge Audit Logs (FS_AUDITLOG_PURGE)

Audit log data can become very large very quickly. As part of the setup process, processes for a rolling purge should be defined. Use the Purge Audit Logs page (FS_AUDITLOG_PURGE) to delete selected audit logs.

PeopleSoft Financials Audit Sources		
Application	**Audit Log Record**	**Transaction Flows**
Asset Management	AM_ASST_AUD_TBL	Asset Adds and Copy Adjustments and Transfers Depreciation Interunit Transfers Recategorizations Retirements and Reinstatements Revaluation
Billing	BI_IVC_AUD_TBL	Create and Edit Billing Invoice Online Copy and Adjust Billing Invoice Correct Budget Stage Error Finalize Billing Invoice Create Installment Invoice Create Recurring Invoice Interface Create/Edit Invoice Billing Invoice Maintenance Approve/Delete Worksheet
General Ledger	GL_AUD_JRNL	Create, Edit and Post Journal Delete Journal Mark to Post and Unpost Journal Unpost Journal Update Journal Unmark to Post and Unpost Journal Journal Date Change
Payables	AP_VCHR_AUD_TBL AP_PYMT_AUD_TBL AP_CNTL_GRP_TBL	Voucher transactions Payment transactions Control Group transactions
Receivables	AR_AUD_DEPOSIT AR_AUD_DRAFT AR_AUD_ITEM AR_AUD_PND_ITEM AR_AUD_PYMNT	Items Drafts Payments Deposits

Application Logging (LogFence)

Log Fence is part of PeopleTools. It allows for application error messages to be consolidated and is set in the application server configuration file (PSSYSADMRV.CFG.) The log can consolidate SQL, application traces along with PeopleTools actions. The logs are written to: PS_CFG_HOME/appserv/prcs/<Database Name>/LOGS which can be loaded into your SIEM.

Verification:

1. On the application server configuration file look in *PS_CFG_HOME*\SYSADMerv*domain_name* for the file PSSYSADMRV.CFG
2. Locate the section: General settings for PSTOOLS
3. Set AppLogFence (see table below).
4. Specify the Log/Output Directory variable in the configuration file to set a common log and output directory. The default is: Log/Output Directory=%PS_SERVDIR%\log_output
5. The default is three (3). For level 3 logging and above, all detailed messages created on the analytic server will be logged both in the application server as well as in analytic server log file
6. For level 4 logging or above, all tracing information is as well logged to the analytic server log file.
7. Be sure to set the corresponding purge rotation settings not to fill the file system (e.g. Recycle Count and Dynamic Change). Note however that dynamic recycling is not recommended for production environments.

Log Fence Settings	
Level	**Description**
-100	Suppress logging
-1	Protocol and memory errors
0	Status information only
1	General errors
2	Warnings
3	Tracing Level 1 – Default.
4	Tracing Level 2
5	Tracing Level 3

Additional PeopleTools Audit Records

Expand the number of PeopleSoft security sensitive tables being audited from Appendix A.

PeopleSoft History Tables

The PeopleSoft Data Archive Manager allows for the creation of history tables where production data can be moved and held. For key security sensitive tables, for example, configuration and setup tables, history records can be loaded into the SIEM for advanced correlation and alerting.

Navigation auditing

With level 1 and two auditing in place, adding navigation auditing to security sensitive forms and records is a logical next step. This is especially useful for monitoring who is viewing sensitive information such as bank account and credit card data etc..... To enable this functionality refer to the following Oracle support whitepaper: PeopleSoft Security Auditing (Doc ID 1963774.1). Once enabled, pull the logs into the SIEM.

Secure Sensitive Data

A large part of the focus for Level 3 should be the logging and auditing DBAs and developers (or anyone else) reading or changing sensitive data using direct database connections outside the PeopleSoft Application.

PeopleTools supports the use of Oracle Fine Grained Auditing (FGA). FGA is a standard (free) feature of the Oracle RDMBS Enterprise Edition. With FGA, audit policies can be created to log specific tables and columns for specific DML events (SELECT, INSERT, UPDATE, DELETE), but only when specific criteria are met – such as when such activity is occurring outside the PeopleSoft application. Refer to Level 1 for a full description of FGA. For clients not using the Oracle RDBMS to support their PeopleSoft implementation, both DB2 and MS SQL-Server offer their variations of FGA.

FGA will log reads (SELECTS) of sensitive tables such as those with Personally Identifiable Information (PII) such as social security numbers and is an idea use case of FGA. FGA policies, since they are written against the base table, are also fired whenever a view containing the targeted table/column is referenced.

Three steps are required to secure sensitive data:

1. Inventory tables with sensitive data
2. Create logon trigger
3. Create FGA policies

The first step to start using FGA would be to inventory the sensitive data within the database, inclusive of standard PeopleSoft tables as well as backup and 'old' tables. Ideally, a cleanup effort will then follow to purge such tables of rouge sensitive data.

The second step for using FGA with PeopleSoft will be to create a database logon trigger to set a context variable. This is a standard recommended best practice for using FGA and allows database sessions to be identified (flagged) as either coming from the PeopleSoft application itself (e.g. GUI forms, SQR, Cobol) or developers and DBAs directly accessing the database from their laptops.

The third step is then to create FGA policies against specific tables and columns of sensitive data. These policies should be written to be fired only when the context variable identifies the database session as NOT coming from the PeopleSoft application.

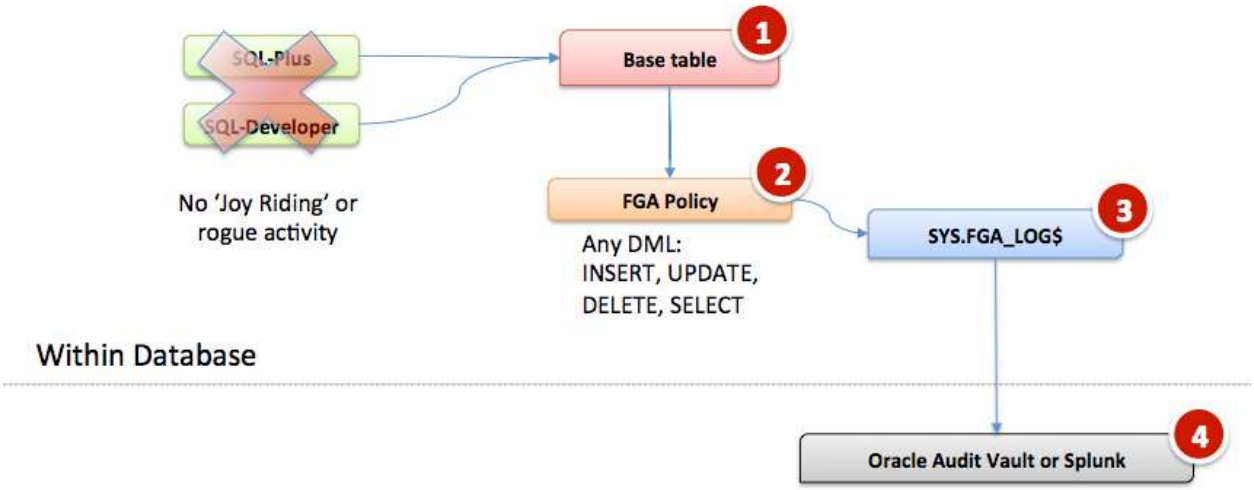


Figure 10 - Protect Sensitive Data

Inventory Tables with Sensitive Data

What tables contain PII data in PeopleSoft? Besides the technical documentation, other sources need to be considered such as staging tables left over from implementation, interfaces, and backups of tables created by DBAs before running data fixes.

The recommended approach is to consult both the PeopleSoft and Oracle RDBMS data dictionaries to first inventory sensitive information at a table column basis. This effort then needs to be followed by a 'blind crawl' of the database using Regular Expressions – click here for more information on Regular Expressions: https://en.wikipedia.org/wiki/Regular_expression.

An example of a Regular Express used to locate a United States Social Security Number is below:

```
^([1-6]\d{2}|0[1-9]\d|00[1-9])([-_\. \|]?)([1-9]\d|\d[1-9])\2([1-9]\d{3}|\d{3}[1-9])$
```

The table below is by no means complete. It is intended only to show there are a good number of columns that hold sensitive data as the result of reviewing both the data dictionaries and blind crawling using regular expressions. Bear in mind too that once initially complete, the PII inventory process will need to be done periodically to ensure the inventory is current.

Example Sensitive Columns	
Table	**Column**
PS_VNDR_BANK_ACCT	BANK_ACCOUNT_NUM
PS_GPAR_GARN_DTL	BANK_ACCOUNT_NUM
PS_SJT_PERSON	NATIONAL_ID
PS_PERS_NID	NATIONAL_ID
PS_EMPLOYEES	NATIONAL_ID
PS_HGA_PG_EMP_DET	NATIONAL_ID
PS_DEP_BENEF_NID	NATIONAL_ID
PS_PAY_CHECK	SSN
PS_W2_DATA	SSN

To fully define a listing of sensitive tables it is highly suggested to use an automated tool. Integrigy Corporation can also assist with this process.

Example FGA Policy

Below is an example of an FGA policy to log developers and/or DBAs making changes to and/or reading (SELECT) the column SSN 'social security number' on the table PS_PAY_CHECK. Note that the policy could easily be created only for SELECTs and/or any combination of DML operations. Also, note that the audit condition is calling a function. This function checks the context variable for the session to determine if the session is coming from the PeopleSoft Application or a direct database connection. In this example, the function is designed to ignore and filter out PeopleSoft application activity to log only those sessions created by developers, DBAs and/or anyone else logging on to the database directly.

```
DBMS_FGA.ADD_POLICY (object_schema      =>  'SYSADM',
       object_name         =>  'PS_PAY_CHECK',
       policy_name         => 'XXXX_FGA_NOT_GUI_PAY_CHECK',
       audit_condition     => 'XXXX_FGA.XXXX_FGA_UTIL.SFUNC_PS_FGA_POLICY_CALLER > 0',
       audit_column        =>  'SSN',
       handler_schema      =>  NULL,
       handler_module      =>  NULL,
       enable              =>  TRUE,
       statement_types     =>  'SELECT, INSERT, DELETE, UPDATE',
       audit_trail         =>  DBMS_FGA.DB,
       audit_column_opts   =>   DBMS_FGA.ANY_COLUMNS);
```

APPENDIX A – RECOMMENDATIONS FOR PEOPLESOFT AUDITING

The following table identifies the records and tables that define security constructs within PeopleSoft that should be considered for PeopleSoft database auditing. Do not attempt to audit all of them. Select those believed appropriate for your specific needs. The intention is to ensure the integrity of the security rules and ensure that no unauthorized changes are made through direct database connections.

Level	Framework	Record	DB Table	RECDESCR
1	E12		AUDIT_CNTRL_TBL	Defines auditing for PeopleSoft Financials
1	E14	PRCSDEFN	PS_PRCSDEFN	Process Defn
1	E7, E8	PSAUTHITEM	PSAUTHITEM	Authorized Menu Item
1	E14	PSCLASSDEFN	PSCLASSDEFN	Permissions Lists Definition
1	E14	PSMENUDEFN	PSMENUDEFN	Menu Definition
1	E13	PSMENUITEM	PSMENUITEM	Menu Item
1	E14	PSMSGNODEDEFN	PSMSGNODEDEFN	Message Node Definition
1	E14	PSOPROBJ	PSOPROBJ	Operator Object Group
1	E12, E13	PSRECDEFN	PSRECDEFN	Record Definition
1	E12	PSRECFIELD	PSRECFIELD	Field definition
1	E7, E8	PSROLECLASS	PSROLECLASS	Role Classes
1	E7, E8	PSROLEDEFN	PSROLEDEFN	Role Definition
1	E9	PSROLEUSER	PSROLEUSER	Role User
1	E4, E5, E6, E14	PSSECOPTIONS	PSSECOPTIONS	Password controls
1	E14	PSSQLTEXTDEFN	PSSQLTEXTDEFN	SQL Object Text
1	E12	PSTRIGGERDEFN	PSTRIGGERDEFN	Defined database triggers
1	E14	PSWEBPROFILE	PSWEBPROFILE	Web Profile
1	E9	PSROLEUSER	PSROLEUSER	User Roles
1	E4, E5	PSOPRDEFN	PSOPRDEFN	User definition
1	E1, E3	PSPTLOGINAUDIT	PSPTLOGINAUDIT	Login history
2	E14	SCRTY_ACC_GRP	PS_SCRTY_ACC_GRP	Access Group Security
2	E14	SCRTY_QUERY	PS_SCRTY_QUERY	PS/Query Profile
2	E14	SCRTY_SET_TBL	PS_SCRTY_SET_TBL	Security Set
2	E14	SCRTY_SRCHGRP	PS_SCRTY_SRCHGRP	Search Group Authorizations
2	E14	SCRTY_TBL_DEPT	PS_SCRTY_TBL_DEPT	OprID Access to Departments
2	E14	SCRTY_TBL_INST	PS_SCRTY_TBL_INST	OprID Access to Institutions
2	E14	SDK_SCRTY_DEPT	PS_SDK_SCRTY_DEPT	SDK User Access to Departments
3	E14	BUS_UNIT_OPT_HR	PS_BUS_UNIT_OPT_HR	Business Unit Options for HR
3	E14	EP_INSTALLATION	PS_EP_INSTALLATION	ePerformance Management
3	E14	GP_INSTALLATION	PS_GP_INSTALLATION	GP Installation
3	E14	GPSINSTALLATION	PS_GPSINSTALLATION	GPS ID get table
3	E14	HRSINSTALLATION	PS_HRSINSTALLATION	eRecruit Installation Table

Level	Framework	Record	DB Table	RECDESCR
3	E14	INSTALLATION	PS_INSTALLATION	Site-Specific Install Options
3	E14	INSTALLATION_AA	PS_INSTALLATION_AA	AA Installation Table
3	E14	INSTALLATION_AD	PS_INSTALLATION_AD	AD Installation Table
3	E14	INSTALLATION_AV	PS_INSTALLATION_AV	Advancement Installation Table
3	E14	INSTALLATION_BN	PS_INSTALLATION_BN	Benefits Installation table
3	E14	INSTALLATION_CC	PS_INSTALLATION_CC	CC Installation Table
3	E14	INSTALLATION_FA	PS_INSTALLATION_FA	Financial Aid Installation Tbl
3	E14	INSTALLATION_FS	PS_INSTALLATION_FS	System Options - PS/Financials
3	E14	INSTALLATION_HR	PS_INSTALLATION_HR	HR Installation Record
3	E14	INSTALLATION_PY	PS_INSTALLATION_PY	PNA Installation table
3	E14	INSTALLATION_SA	PS_INSTALLATION_SA	Student Admin Install Options
3	E14	INSTALLATION_SR	PS_INSTALLATION_SR	
3	E14	JPMINSTALLATION	PS_JPMINSTALLATION	JPM Installation Table
3	E14	APPR_RULE_HDR	PS_APPR_RULE_HDR	Approval Rule Defn Hdr
3	E14	PSURLDEFN	PSURLDEFN	URL Table
3	E14	SJT_PERSON	PS_SJT_PERSON	Security dat for Person Access
3	E14	PSCRYPTDLLDEFN	PSCRYPTDLLDEFN	Encryption Libraries
3	E14	PSCRYPTKEYSET	PSCRYPTKEYSET	Encryption Libraries
3	E14	PSCRYPTPRFL	PSCRYPTPRFL	Encryption Libraries
3	E14	PSFILEREDEFN	PSFILEREDEFN	Libraries registered
3	E14	PSIBPROFILE	PSIBPROFILE	IB system settings.
3	E14	PSOPERATION	PSOPERATION	IB Services
3	E14	PSOPTIONS	PSOPTIONS	PeopleTools System Options
3	E1	PSPRDMDEFN	PSPRDMDEFN	Portal Definition Table
3	E13		PSRECFIELDALL	Field definition
3	E14	PSSEC_PPC_OPTN	PSSEC_PPC_OPTN	Defines PeopleCode Options
3	E14	PSSTATUS	PSSTATUS	PeopleTools System Control
3	E14	PSTREEDEFN	PSTREEDEFN	Tree Definition
3	E14	MAINTENANCE_LOG	PS_MAINTENANCE_LOG	Patch history
3	E14	TL_INSTALLATION	PS_TL_INSTALLATION	Installation Time & Labor Tbl
3	E14	US_INSTALLATION	PS_US_INSTALLATION	Installation Table USA
3	E14	PSBUSPROCDEFN	PSBUSPROCDEFN	Business Process Definition
3	E1, E2	PSACCESSLOG	PSACCESSLOG	Login history
3	E14	PSSERVERSTAT	PSSERVERSTAT	Process Server Statistics
3	E13	PSPNLDEFN	PSPNLDEFN	Panel Definition

APPENDIX B – PEOPLESOFT TRIGGERS AND SHADOW AUDIT TABLES

This whitepaper assumes the use of an Oracle RDBMS. While not recommended, primarily because of the performance overhead and complexity to implement, it is possible to use PeopleSoft's database agnostic logging and monitoring solution that utilizes database triggers and shadow tables. This appendix describes how to enable auditing by generating triggers and shadow tables instead of using the recommended Fine Grained Auditing features of Oracle, DB2, and MS SQL-Server.

PeopleSoft's default auditing send record and field level data to centralized tables (PSAUDIT). Field level auditing has several disadvantages. First, field level auditing secures only activity that occurs within the PeopleSoft user interface. Direct database activity from DBAs and developers using SQL-Plus and/or SQL-Developer will not be detected by field level auditing, nor will field level auditing secure activity occurring within SQR and PeopleSoft's COBOL programs. Most importantly, however, field level auditing does not support auditing of PeopleTools tables (PT 8.1x, 8.4x and PT 8.5x). Consequently, Integrigy's Log and Audit Framework for PeopleSoft uses the database-auditing alternative.

A key point for Level 1 is to enable auditing for changes to security sensitive database (INSERTS, UPDATES, and DELETEs), not reads (SELECTS) of sensitive information. The difference being the goal of level 1 is to detect unauthorized changes to security rules vs. potential abuse by developers and DBAs to read sensitive data (e.g. salary data and/or social security numbers). Level 3 of the Framework being presented here makes recommendations for how to log and audit reads of sensitive data.

Implementing database trigger-based auditing will create a performance impact, however, the Integrigy Framework for PeopleSoft Logging and Auditing targets high-security impact, low volume transactions so as to alleviate any potential security impact.

The following steps are detailed below to enable database auditing within PeopleSoft:

1. Enable Login Auditing
2. Enable DB Monitoring
3. Create PS_ORID function
4. Verify Existing Audit Triggers
5. Define Audit records (shadow tables)
6. Define Audit Triggers
7. Deploy Audit Triggers
8. Setup Rolling Purge of Audit Tables

Enable Login Auditing

This step is not part of database auditing but is good to do before enabling database auditing.

Enable the Login Audit option use PSADMIN (psSYSADMrv.cfg) to ensure the following domain parameters are set for auditing. This will log User logon/off and attempts to the table PSPTLOGINAUDIT

On the application server configuration file look in *PS_CFG_HOME*\SYSADM*domain_name* for the file psappsrv.cfg. Locate the parameter 'Enable Login Audit' and Set Enable Login Audit option = Y

Enable DB Monitoring

First, use PSADMIN (psappsrv.cfg) to ensure the domain parameters are set for auditing to allow database triggers to tables with the 'AUDIT_prefix'.

Verification:

1. On the application server configuration file look in *PS_CFG_HOME*\SYSADM*domain_name* for the file psappsrv.cfg (for example in /home/psadm2/psft/pt/8.54/appserv/APPDOM/psappsrv.cfg)
2. Locate the parameter EnableDBMonitoring
3. Make user EnableDBMonitoring = 1

Create PS_OPRID Function

For Oracle audit triggers, for the audit triggers to obtain the PS_OPRID, PeopleSoft provides a function. This function must be installed into the Oracle database schema for the PeopleSoft database before creating the audit triggers. This function is installed by executing the following SQL as the PeopleSoft database owner ID:

$PS_HOME\scripts\getpsoprid.sql

For other database, platforms refer to the documentation[3].

Verify Existing Audit Triggers

Verify what database trigger based auditing has already been enabled:

Verification:

-- List sensitive tables with database trigger auditing enabled

```
SELECT PSRECDEFN.RECNAME , PSRECDEFN.SQLTABLENAME,
NVL(TRIM(PSRECDEFN.SQLTABLENAME),'PS_'||PSRECDEFN.RECNAME) THETABLE ,
PSRECDEFN.OBJECTOWNERID,
PSRECDEFN.FIELDCOUNT,
PSRECDEFN.RECDESCR,
PSRECDEFN.DESCRLONG,
OPTTRIGFLAG,
SYSTEMIDFIELDNAME,
TIMESTAMPFIELDNAME,
PSTRIGGERDEFN.*
FROM SYSADM.PSTRIGGERDEFN , SYSADM.PSRECDEFN
WHERE PSRECDEFN.RECNAME = PSTRIGGERDEFN.RECNAME
ORDER BY 1,3;
```

Define Audit Records

To create the shadow tables for database auditing you need to the client/server Application Designer Tool. Configure records for database trigger auditing. This process must be done per the documentation. ***Do not manually create the shadow audit tables and/or triggers***. Once created, DataMover can be used to migrate the audit records and triggers among non-production instances and/or to production.

Refer to Appendix A for the full listing. Below is a recommended short list to start with. Use the Status column to record your progress configuring the audit records.

---

3 https://docs.oracle.com/cd/E58500_01/pt854pbh1/eng/pt/tadm/concept_UnderstandingDatabaseLevelAuditing-077a5f.html - topofpage

Recommended Level 1 Auditing					
No	**Framework**	**Record**	**DB Table**	**Description**	**Status**
1	E14	PRCSDEFN	PS_PRCSDEFN	Process Definition	
2	E1, E2	PSACCESSLOG	PSACCESSLOG	Login history (only for update or delete)	
3	E14	PSCLASSDEFN	PSCLASSDEFN	Permissions Lists Definition	
4	E14	PSMENUDEFN	PSMENUDEFN	Menu Definition	
5	E13	PSMENUITEM	PSMENUITEM	Menu Item	
6	E4, E5	PSOPRDEFN	PSOPRDEFN	User definition	
7	E1, E3	PSPTLOGINAUDIT	PSPTLOGINAUDIT	Login history (only for update or delete)	
8	E12, E13	PSRECDEFN	PSRECDEFN	Record Definition	
9	E7, E8	PSROLECLASS	PSROLECLASS	Role Classes	
10	E7, E8	PSROLEDEFN	PSROLEDEFN	Role Definition	
11	E9	PSROLEUSER	PSROLEUSER	Role User	
12	E4, E5, E6, E14	PSSECOPTIONS	PSSECOPTIONS	Password controls	
13	E12	PSTRIGGERDEFN	PSTRIGGERDEFN	Defined database triggers	
14	E14	PSWEBPROFILE	PSWEBPROFILE	Web Profile	

Steps to create shadow audit tables:

1. From the client/server Application Designer, open the record definition of the PeopleSoft base table record
2. Create a copy of the base table by saving it as a new record, prefaced with AUDIT_<base table name>.
3. Remove all existing edit and key attributes for the existing columns.
4. Add to the top of the audit record the following three (3) audit-specific fields "columns." These tables must be the first three columns and must be spelled exactly.
5. Each of the three audit columns must be a “Required” field and must be a “Key” field.
6. Remove base columns not needed to support auditing. Select only the minim number of columns required and/or deemed necessary for auditing.
7. Save to commit all changes to the PeopleSoft metadata dictionary for the new table “record” – click the Save icon in the ribbon
8. New the create table use the Build menu in the ribbon. Select the shadow audit table(s) to create in the database. This step will only generate a file in the local Windows operating system where the Application Designer is running.
9. Open the DDL file by double clicking on the name of the file. This will open the DDL file in Notepad. Copy the DDL.
10. Login to SQL-Plus, TOAD or SQL-Developer as SYSADM and past the DDL to create the shadow audit table.

Audit Field Name	Purpose	Data Type

Audit Field Name	Purpose	Data Type
AUDIT_OPRID	Identifies the user who causes the system to trigger the audits, either by performing an add, change, or delete to an audited field.	Character
AUDIT_STAMP	Identifies the date and time that the audit is triggered.	Datetime
AUDIT_ACTN	Indicates the type of action the system audited. Possible action values include: • *A* – Row inserted. • *D* – Row deleted. • *K* – Row updated, snapshot before update. • *N* – Row updated, snapshot after update.	Character

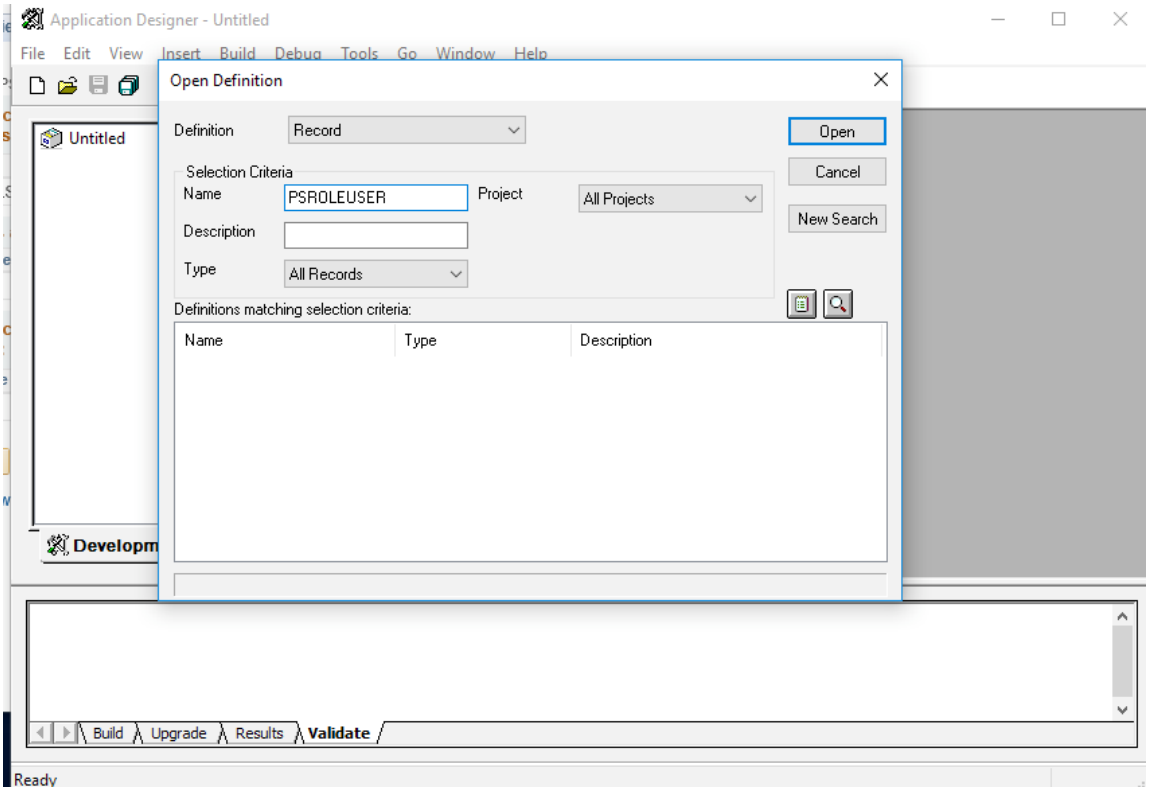


Figure 11 - Locate the base table

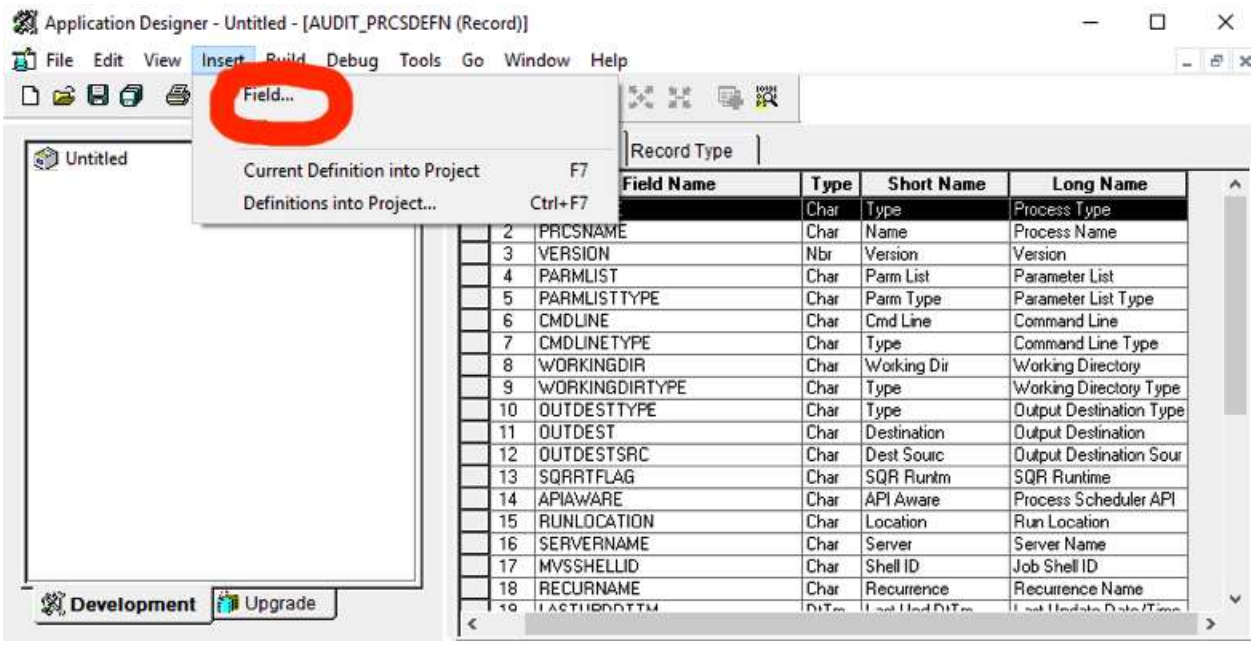


Figure 12 - Insert Field(s)

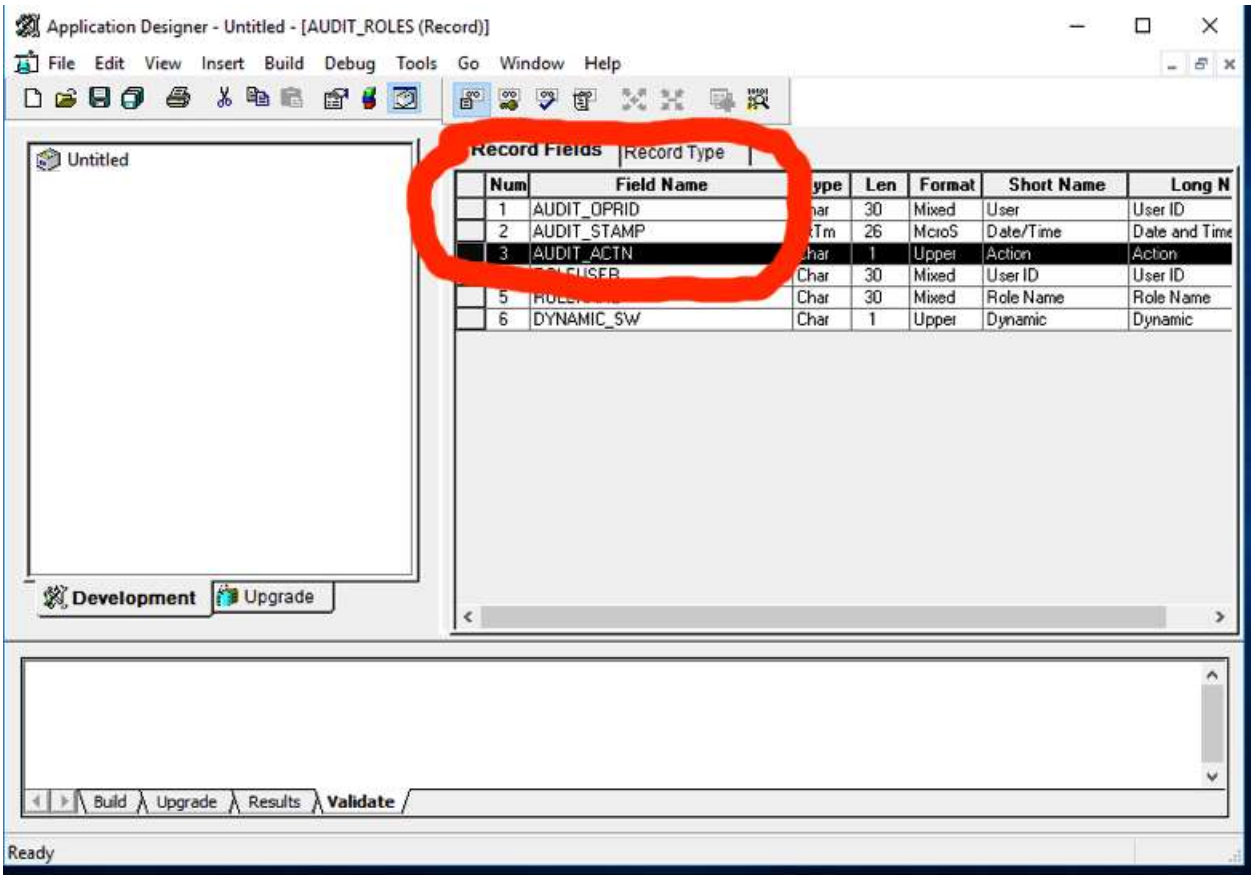


Figure 13 - Create the three audit columns

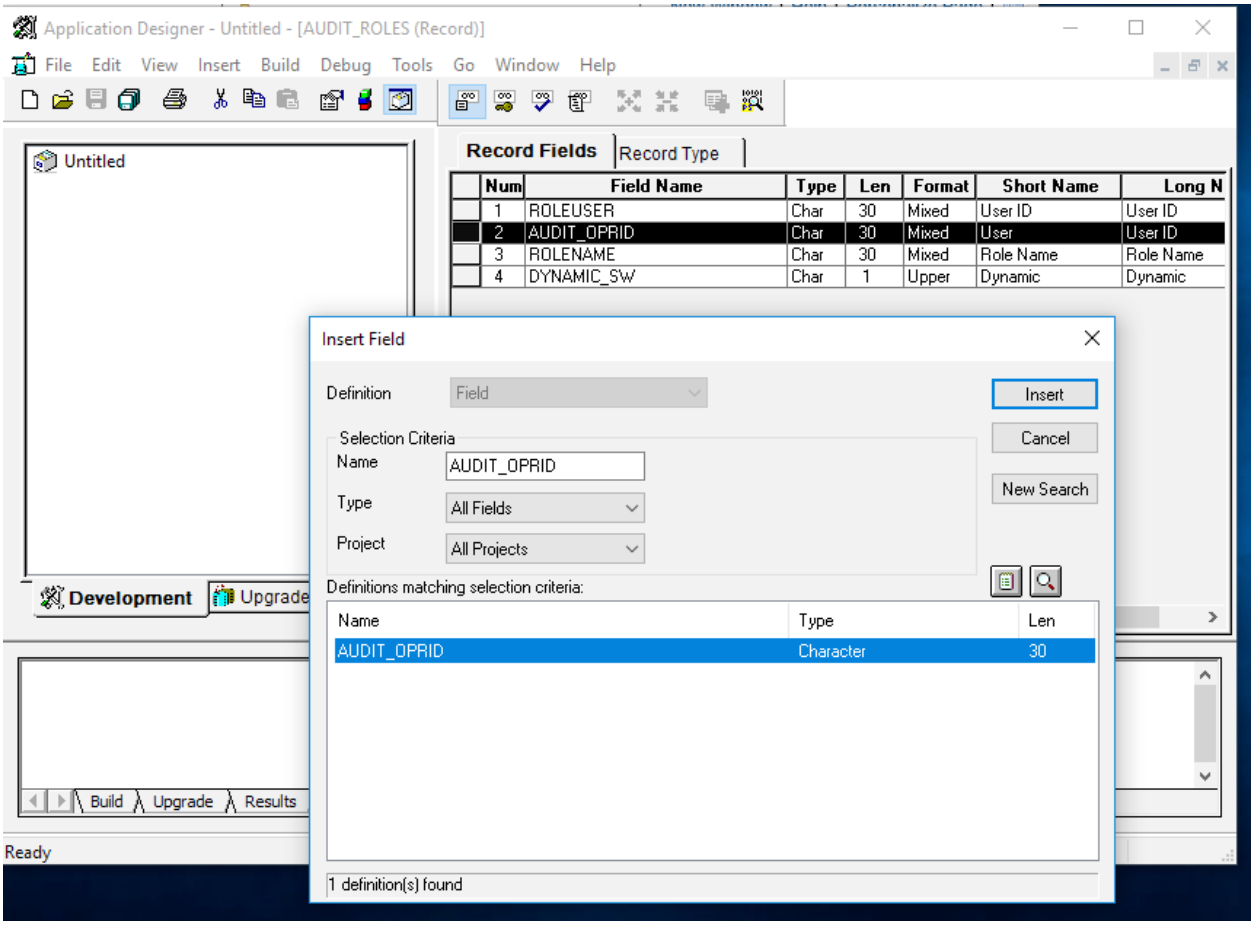


Figure 14 - Example of creating an audit column

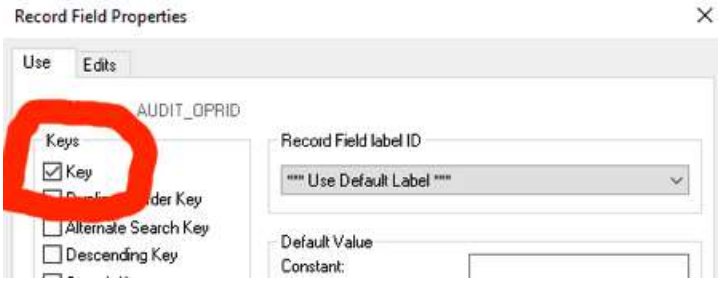


Figure 15 - Each audit column must be a Key Field

Figure 16 Each audit column must be a required field

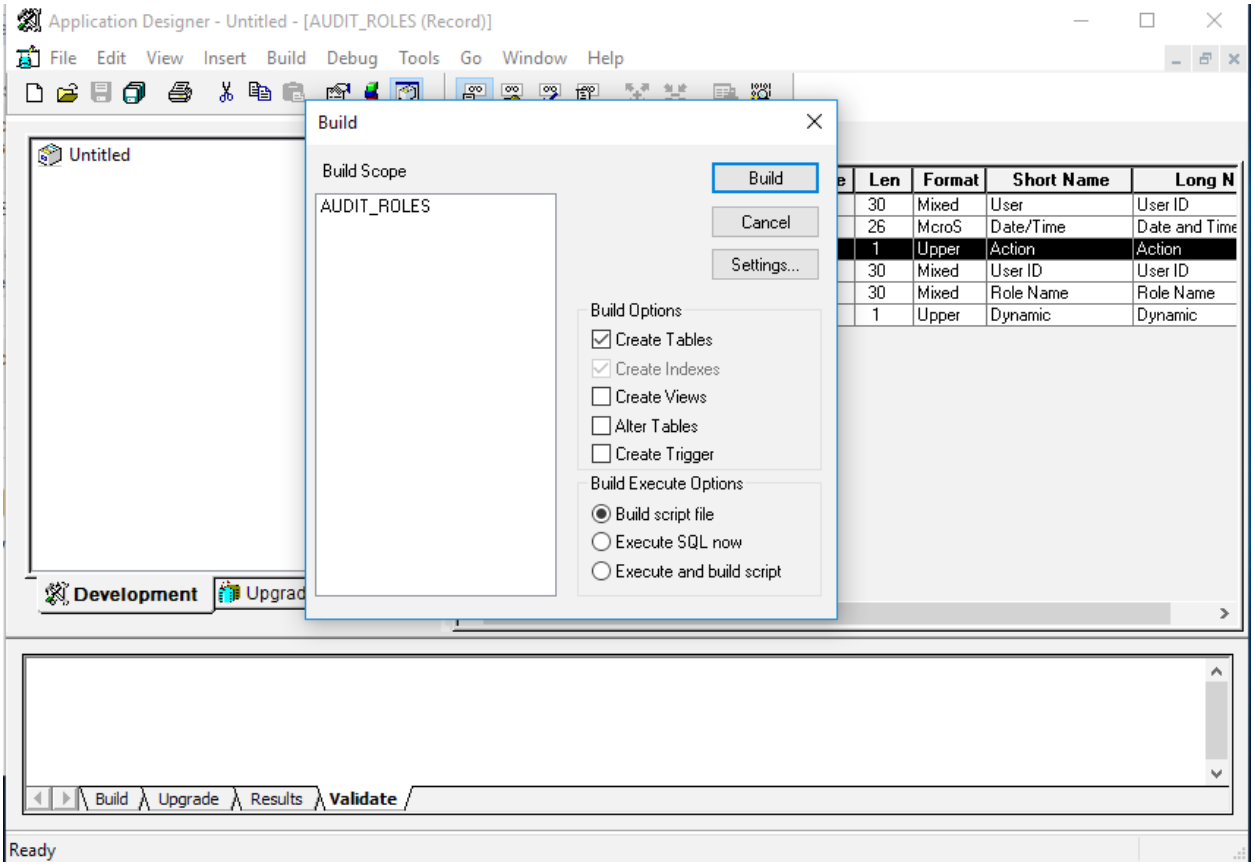


Figure 17 - Generate DDL to build new table

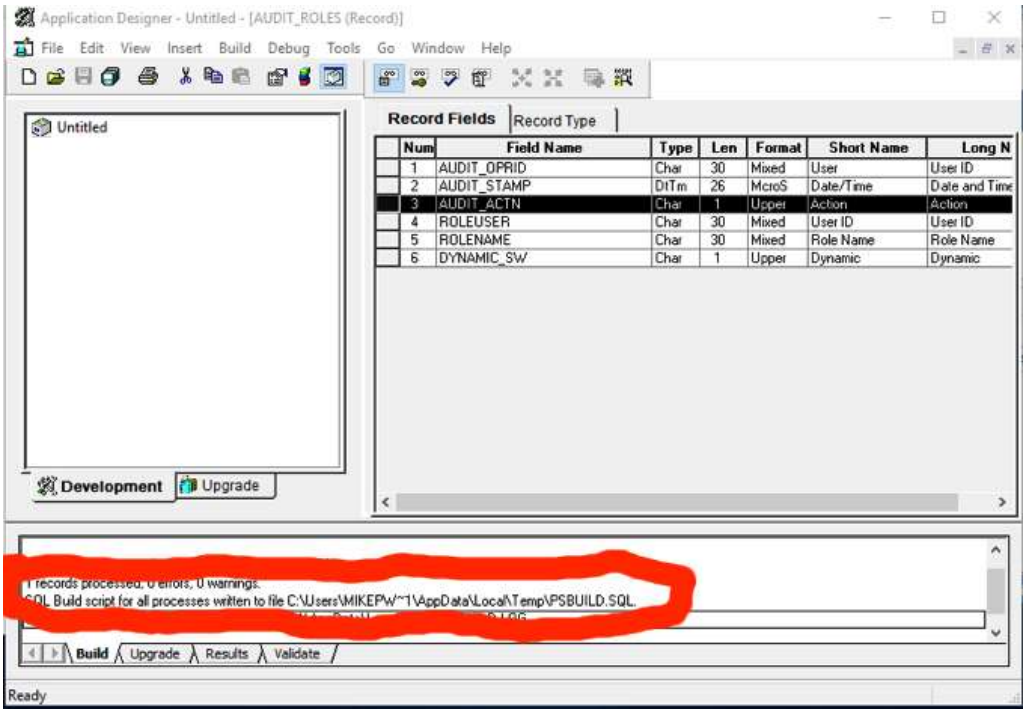


Figure 18 - Double Click on the file name to open the DDL table create script and copy it

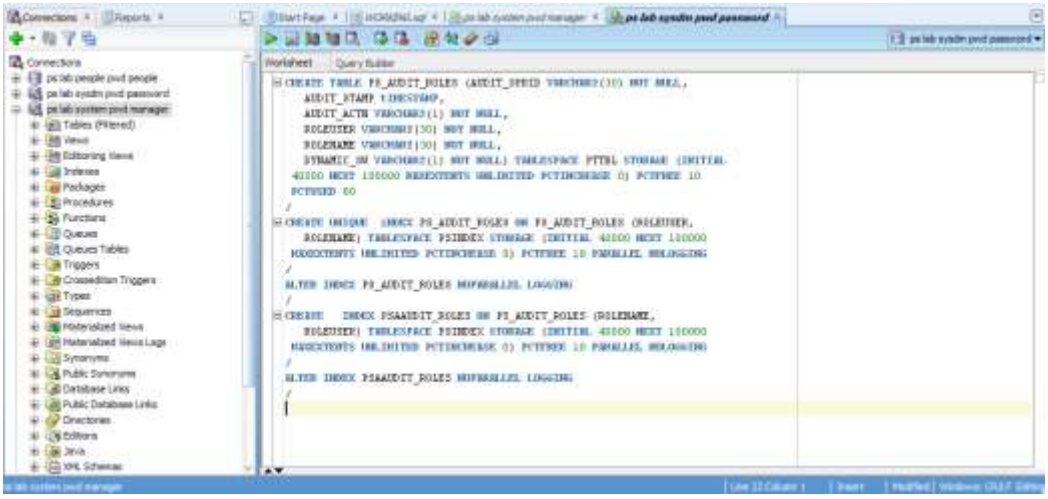
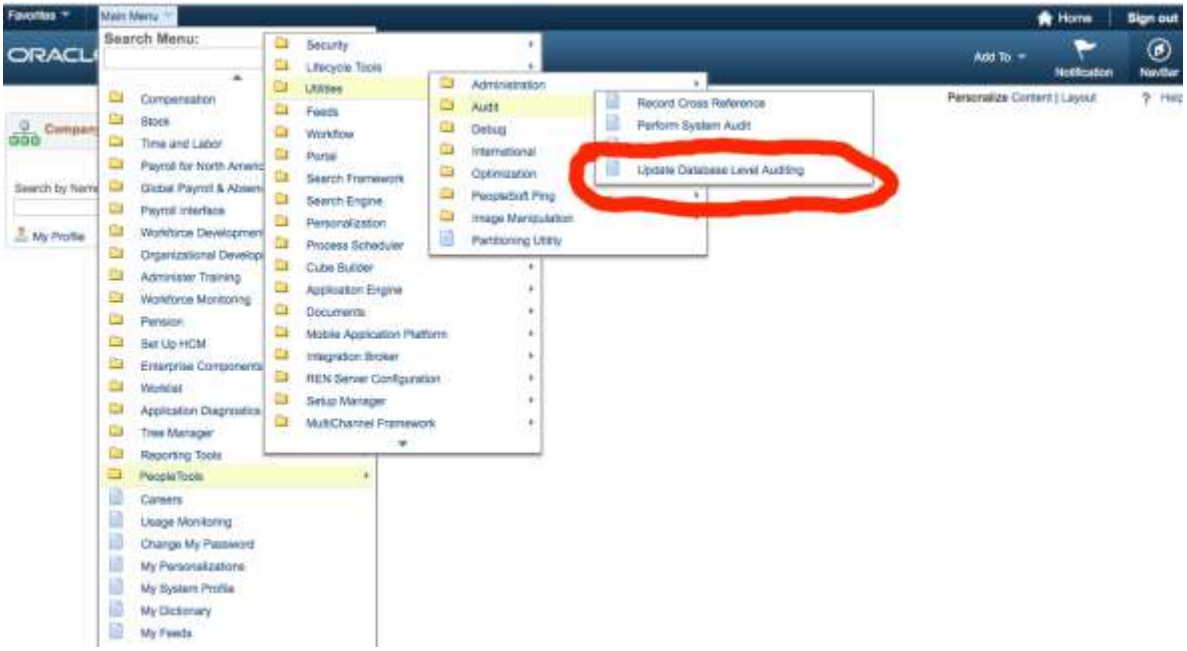


Figure 19 - Past table DDL script and run as SYSADM

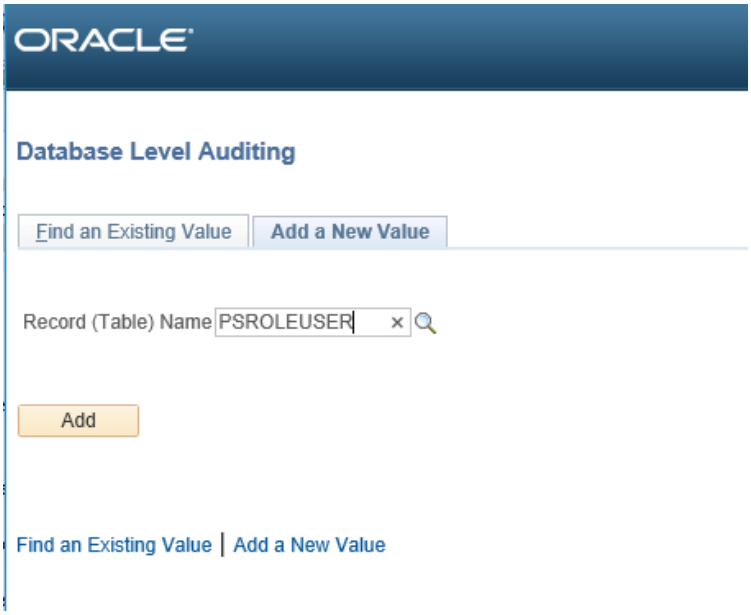
Generate and Deploy Audit Triggers

Once you have created the shadow audit tables both in the PeopleSoft data dictionary and in the database, you need to create the triggers to copy all changes from the base tables to the shadow audit tables.

1. From the PeopleSoft application (not the client/server Application Designer), navigate to PeopleTools -> Utilities -> Audit -> Update Database Level Auditing.



2. Find the shadow audit table “record” and click Add a New Value.



3. Create and audit record. This step will generate the DDL for the trigger and associate the trigger with the shadow audit table "record." Perform the following steps:
    1. Be sure to use the prefix "*AUDIT_*" for the name,
    2. Click the checkboxes for Add, Change and Delete and
    3. Click "Generate Code"
    4. Click Save".

    This step will only generate the SQL DDL to create the trigger, but it will NOT execute the DDL. The 'generate code' button will just deposit the DDL into the table PSTRIGGERDEFN. Once all the DDL for all triggers is generated, the last step will be to run a single consolidated DDL script to generate triggers.

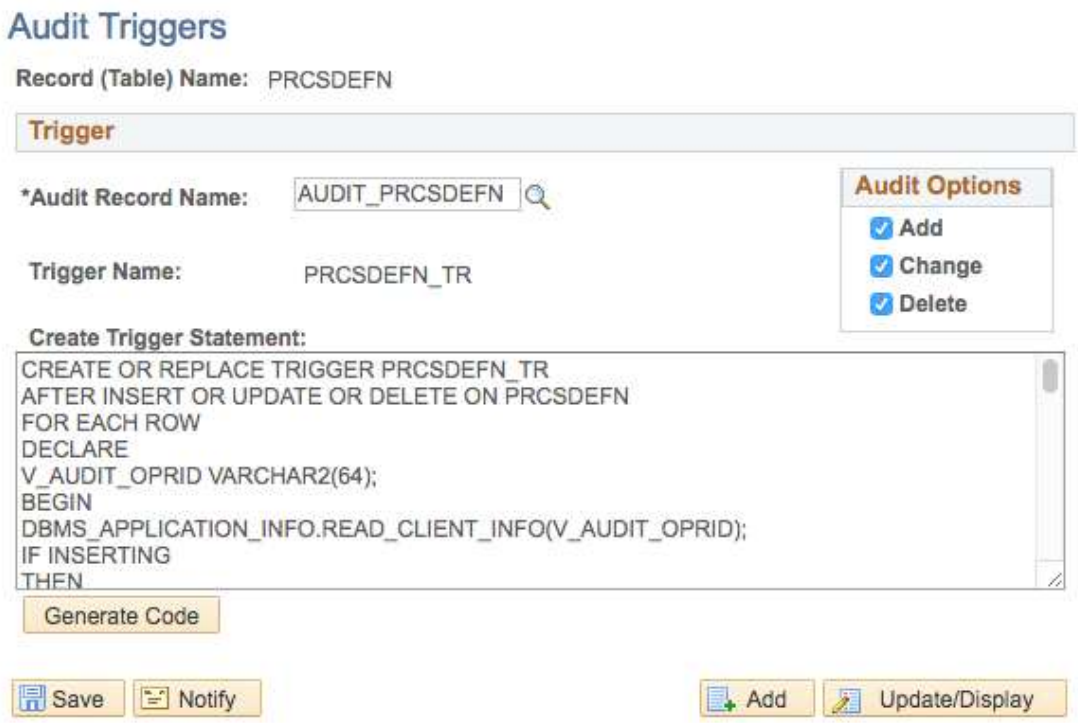


Figure 20 - Example of Trigger Definition - Note AUDIT_ prefix

Deploy Audit Triggers

Once all the base tables "records" have been created the DDL generated to create the triggers, a batch job needs to be run in the Process Scheduler to create a single DDL script to deploy the triggers. Follow the steps below to deploy all the triggers.

1. In the Application , navigate to PeopleTools -> Utilities -> Audit -> Perform Database Level Audit
2. Select the triggers to be included in the DDL deployment script, either select ALL or just those defined in step three above. Then click RUN and note the process ID assigned.

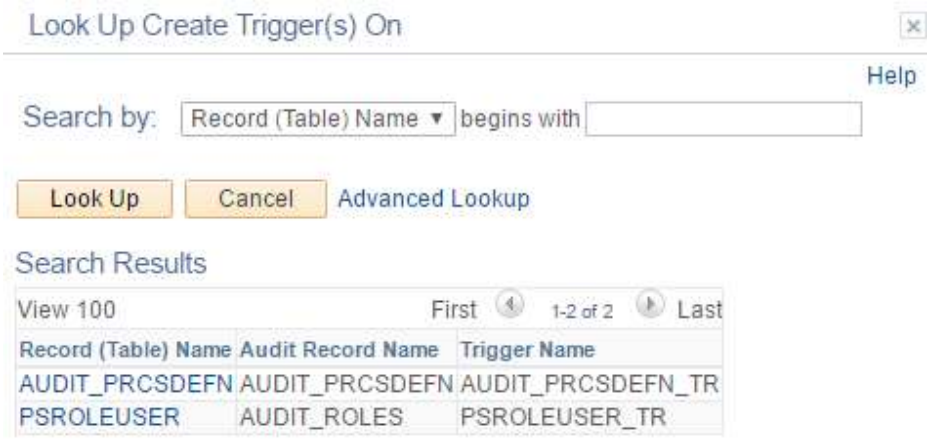


Figure 21 - Select Triggers and run batch job to create consolidated DDL script

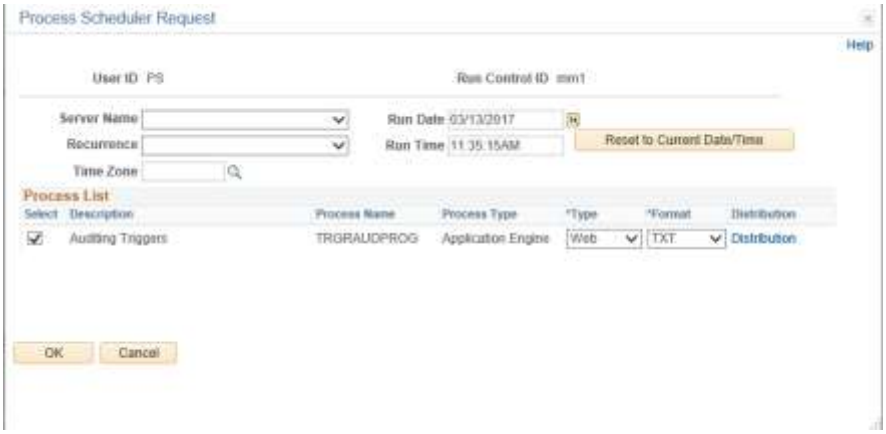


Figure 22 - Run the process

3. Once the SQL script is generated, locate the file in the PS_SRVRDIR directory in the Unix file system where the Process Scheduler is being run. For Windows, the file will be created in the directory the %TEMP% environment variable specifies. The file name will be TRGCODEX.SQL, where X represents a digit that is determined by the number of files by the same name that already exists in the output directory.

   In the demo environment, an example is below:
   ./home/psadm2/psft/pt/8.54/psreports/PSHCM92/20170313/16278/AE_TRGRAUDPROG_63559.stdout

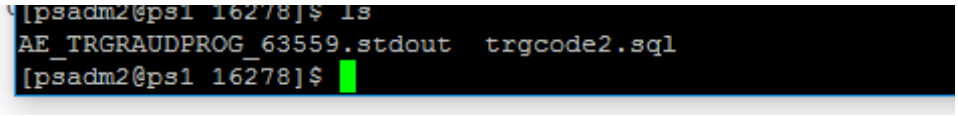


Figure 23 - Find the trgcodeX.sql file

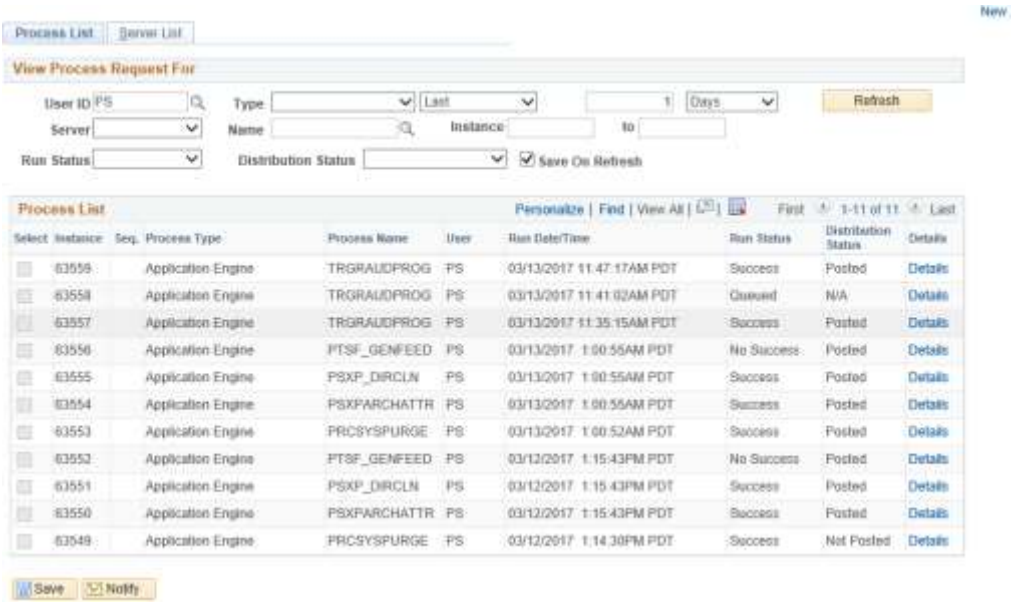


Figure 24 - Monitor Process Manager Job

4. WINSCP and/or copy the file TRGCODEX.sql
5. Open the file TRGCODEX.sql in SQL-Developer using the SYSADM account and run the script.

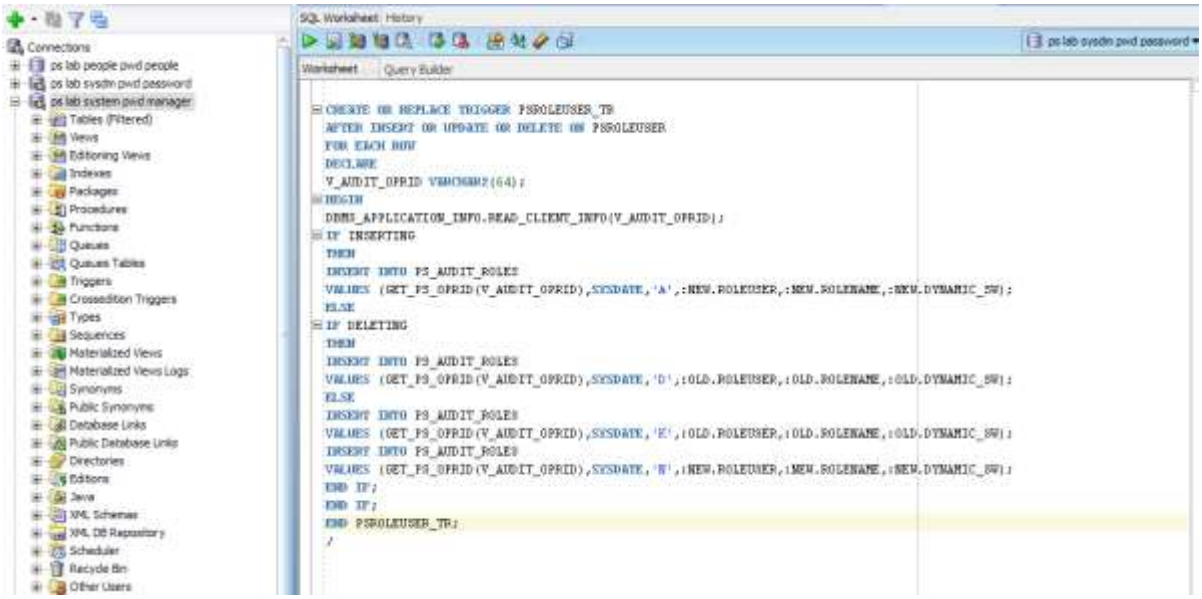


Figure 25 - Example of running trgcodex.sql in SQL-Developer

Secure Shadow Audit Tables

The intent of the PeopleSoft shadow audit tables to provide trust verification. That is to be able to establish a record of all changes to the security sensitive tables. Allowing those whose trust needs to be verified to also be able to modify the audit table negates the purpose of the audit table. To secure the integrity of the audit trails being written to the shadow audit tables, the shadow audit tables themselves need to be audited and monitored.

1. Appropriate to job personnel and staff job functions, restrict access to the accounts and password with access to the shadow audit tables and the sys.aud$ and/or sys.fga_log$ tables.
2. Enable auditing and monitoring on the audit triggers (e.g. AUDIT ALTER TRIGGER). If the audit triggers become disabled, it should be alerted.
3. Use Oracle Fine Grained Auditing (FGA) to audit updates and deletes on the shadow audit tables for accounts other than SYSADM. FGA will need to be created for each shadow table, and ideally, the policies should also incorporate logic to detect SQL NOT coming to the application sever – for example from direct database connection from a laptop using SQL-Developer or SQL-Plus.

Refer to the Oracle RDBMS documentation for more information on Fine Grained Auditing[4]. One detail to note is that unlike traditional Oracle RDBMS auditing, to enable and start to use FGA, no bounce of the database is required.

Setup Rolling Purge of Audit Tables

Step up rolling purge on shadow audit tables per business requirements. Also, be sure to setup rolling purges of both the FGA and traditional Oracle database auditing. Refer to respective Oracle RDBMS and PeopleSoft Archiving documentation.

Oracle provides standard functionality for the rolling purge of SYS.AUD$ and the SYS.FGA_LOG$ tables. Refer to the Oracle RDBMS documentation for more information. For the PeopleSoft shadow audit tables, you will need to manually purge them per your business requirements.

[4] http://docs.oracle.com/database/121/DBSEG/audit_config.htm#DBSEG60681

APPENDIX C – USEFUL SQL

To list what tables are enabled for database trigger auditing:

Defined FGA Policies

```
SELECT POLICY_NAME, ENABLED, OBJECT_SCHEMA, OBJECT_NAME,POLICY_COLUMN,
POLICY_TEXT
FROM SYS.DBA_AUDIT_POLICIES
ORDER BY 1,2;
```

FGA Activity

```
SELECT * FROM DBA_FGA_AUDIT_TRAIL;

SELECT * FROM SYS.FGA_LOG$
ORDER BY NTIMESTAMP# DESC;
```

Triggers Defined for Auditing

```
SELECT * FROM SYSADM.PSTRIGGERDEFN;

-- list tables with auditing triggers
SELECT PSRECDEFN.RECNAME , PSRECDEFN.SQLTABLENAME,
NVL(TRIM(PSRECDEFN.SQLTABLENAME),'PS_'||PSRECDEFN.RECNAME) THETABLE ,
PSRECDEFN.OBJECTOWNERID,
PSRECDEFN.FIELDCOUNT,
PSRECDEFN.RECDESCR,
PSRECDEFN.DESCRLONG,
OPTTRIGFLAG,
SYSTEMIDFIELDNAME,
TIMESTAMPFIELDNAME,
PSTRIGGERDEFN.*
FROM SYSADM.PSTRIGGERDEFN , SYSADM.PSRECDEFN
WHERE PSRECDEFN.RECNAME = PSTRIGGERDEFN.RECNAME;
```

Record auditing

The following SQL identifies records with auditing enabled.

```
SELECT
RECNAME,
RECDESCR,
AUDITRECNAME as TABLE_WHERE_REC_WRITTEN,
CASE WHEN BITAND(RECUSE,1) > 0 THEN 'Y' ELSE 'N' END AUDIT_ADD, CASE WHEN
BITAND(RECUSE,2) > 0 THEN 'Y' ELSE 'N' END AUDIT_CHANGE, CASE WHEN
BITAND(RECUSE,4) > 0 THEN 'Y' ELSE 'N' END AUDIT_DELETE, CASE WHEN
BITAND(RECUSE,8) > 0 THEN 'Y' ELSE 'N' END AUDIT_SELECTIVE
FROM SYSADM.PSRECDEFN
WHERE TRIM(AUDITRECNAME) IS NOT NULL
```

```
ORDER BY RECNAME;
```

Field auditing enabled

The following SQL identifies fields on records that have field level auditing enabled. Field records will be written to the table PSAUDIT.

Verification:

```
SELECT
F.RECNAME,
F.FIELDNUM,
F.FIELDNAME,
F.USEEDIT,
CASE WHEN BITAND(F.USEEDIT,8) > 0 THEN 'Y' ELSE 'N' END AUDIT_FIELD_ADD, CASE
WHEN BITAND(F.USEEDIT,128) > 0 THEN 'Y' ELSE 'N' END AUDIT_FIELD_CHANGE, CASE
WHEN BITAND(F.USEEDIT,1024) > 0 THEN 'Y' ELSE 'N' END AUDIT_FIELD_DELETE
FROM
SYSADM.PSRECFIELD F
WHERE
F.FIELDNAME = (
SELECT
CASE WHEN (
BITAND(USEEDIT,8) > 0 OR BITAND(USEEDIT,128) > 0 OR BITAND(USEEDIT,1024) > 0
) THEN FIELDNAME ELSE '' END AS FIELD_AUDITED FROM SYSADM.PSRECFIELD
WHERE RECNAME = F.RECNAME
AND FIELDNAME = F.FIELDNAME )
ORDER BY F.RECNAME, F.FIELDNUM;
```

REFERENCES

GENERAL

Integrigy Guide to Database Auditing and Logging
https://www.integrigy.com/security-resources/integrigy-guide-database-auditing-and-logging

Security, Audit and Control Features – Oracle PeopleSoft 3rd edition, ISACA,
https://preview.tinyurl.com/lfnperz

PeopleBooks: PeopleTools 8.54: Data Management, Oracle Corporation, November 2016, Chapter Five:
http://docs.oracle.com/cd/E58501_01/psft/pdf/pt854tadm-b1114.pdf

PeopleSoft Security Auditing (Doc ID 1963774.1), Oracle Corporation, January 2015
https://support.oracle.com/rs?type=doc&id=1963774.1

How to Enable PeopleSoft Database Level Auditing (Doc ID 612310.1)
https://support.oracle.com/rs?type=doc&id=612310.1

Preserve FGA polices during PeopleTools upgrades https://preview.tinyurl.com/m2fsbkd

Security, Audit and Control Features – Oracle RDBMS 3rd edition, ISACA,
http://preview.tinyurl.com/mdw5qxd

PeopleSoft for the Oracle DBA, David Kurtz, Apress Publishing
https://www.apress.com/la/book/9781430237075

Integrigy Oracle PeopleSoft Security Quick Reference Guide, Integrigy Corporation, Version 2.0, March 2016
https://www.integrigy.com/security-resources/peoplesoft-security-quick-reference

ABOUT INTEGRIGY

Integrigy Corporation (www.integrigy.com)

Integrigy Corporation is a leader in application security for enterprise mission-critical applications. AppSentry, our application, and database security assessment tool assists companies in securing their largest and most important applications through detailed security audits and actionable recommendations. AppDefend, our enterprise web application firewall is specifically designed for PeopleSoft. Integrigy Consulting offers comprehensive security assessment services for leading databases and ERP applications, enabling companies to leverage our in-depth knowledge of this significant threat to business operations.

INTEGRIGY
Integrigy Corporation
P.O. Box 81545
Chicago, Illinois 60681 USA
888/542-4802
www.integrigy.com