DB Audit Expert™ 4.2

User's Guide

Supported database systems:

- Oracle 7.3, 8.0, 8i, 9i, 10g, 11g
- Microsoft SQL Server 7, 2000, 2005, 2008, 2012
- Sybase SQL Server and Sybase Adaptive Server Enterprise 10.x, 11.x, 12.x, 15.x
- Sybase Adaptive Server Anywhere 6, 7, 8, 9, 10
- IBM DB2 7.x, 8.x, 9.x, 10.x for Linux, Unix, and Windows
- IBM DB2 6.x, 7.x, 8.x, 9.x for z/OS and OS/390
- IBM DB2 5.x for OS/400
- MySQL 4.2, 5.x for Windows and Linux

Contents

About This Guide

This manual describes the features of the DB Audit Expert product, including how to use DB Audit graphical user interface, the audit functions and reports independent of any specific database application. Unless otherwise noted, features and how-to instructions described in this manual apply to all supported database management systems running on any platform.

Intended Audience

This document is intended for Database Administrators, Database Managers, Security Officers, System Administrators and Database Owners.

Conventions used in this document

This section describes the style conventions used in this document.

Italic

An *italic* font is used for filenames or filepaths, URLs, emphasized text, and the first usage of technical terms.

`Monospace`

A `monospaced` font is used for code fragments and data elements.

Bold

A **bold** font is used for important messages, names of options, names of controls and menu items, and keys.

User Input

Keys are rendered in **bold** to stand out from other text. Key combinations that are meant to be typed simultaneously are rendered with "+" sign between the keys, such as:

Ctrl+F

Keys that are meant to be typed in sequence will be separated with commas, for example:

Alt+S, H

This would mean that the user is expected to type the Alt and S keys simultaneously and then to type the H key.

Graphical symbols

 **-** This symbol is used to indicate DBMS specific options and issues and to mark useful auditing tips.

 **-** This symbol is used to indicate important notes.

Abbreviations and Product Reference Terms

DBMS – Database Management System

Oracle – This refers to all supported Oracle® database servers

SQL Server – This refers to all versions of Microsoft® SQL Server™ database servers.

ASE – This refers to all versions of the Sybase® SQL Server™ and Sybase® Adaptive Server® Enterprise database servers.

ASA – This refers to all versions of the Sybase® Adaptive Server® Anywhere database servers.

DB2 – This refers to all versions of the IBM® DB2® database servers.

MySQL – This refers to all versions of the MySQL database servers.

The terms ‘DB Audit Expert’ and ‘DB Audit’ are used interchangeably in this document – they both refer to the same product]

Trademarks

reserved.

CHAPTER 1: Overview of DB Audit Expert

Introduction

Relational Database Management Systems (DBMS) are used today as the primary repository for storing enterprise's most valuable information. In recent years, the volume of data stored in these repositories has grown rapidly. Maintaining fast and easy access to data remains a top priority, while protecting data is becoming more and more difficult. The number of attacks reported against enterprise data systems more than tripled in 2007 alone, and the number continues to rise rapidly. With attacks becoming more sophisticated, faster, and more aggressive, there is growing emphasis on consistent assessment of database vulnerabilities, preventative measures, and database security management.

DB Audit Expert is a professional database security assessment, auditing and security solution that supports a number of widely used Database Management Systems. DB Audit Expert allows database and system administrators, security administrators, auditors, and operators to properly protect and secure their database systems, as well as track and analyze any database activity including database access and usage, data creation, and data changes and deletions.

DB Audit is the only multi-platform multi-database solution covering the full spectrum of database security controls:

- Preventive security management
- Detective and forensic analysis of effective security settings
- Auditing, monitoring and compliance
- Vulnerabilities and penetration testing
- Corrective actions

Key Benefits

- Improves system security and ensures system accountability
- Features centralized security management and auditing of multiple database systems from a single location
- Features easy to use unified security management graphical interface that shortens the learning curve
- Provides audit trail details and reports that are unavailable from a native database audit utility
- Creates analytical reports that reduce large amounts of audit data to meaningful information for identifying various database security violations and misuses
- Generates real-time email alerts and sends them to key personnel when changes occur to sensitive data
- Frees DBAs from the need to create and manage finely tuned database triggers for data-change auditing purposes
- Provides totally transparent auditing of any existing applications without requiring that changes be made to those applications.

Feature matrix by DBMS

DBMS Name and Version	Oracle				Microsoft SQL Server			Sybase Adaptive Server Enterprise				Sybase Adaptive Server Anywhere			IBM DB2			MySQL	
Function	7.3	8.0	8i	9i - 11g	6.5	7	2000 - 2008	10.x	11.0 - 11.1	11.5 - 11.9	12.x - 15.x	6.x	7.x	8.x – 9.x	5.x	6.x	7.x- 9.x	4.2	5.x
System Audit	X	X	X	X	N/A		X		X[4]	X	X	N/A	N/A	N/A		X[5]	X[5]	X	X
Data-Change Audit	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	N/A	X
Data-Change Audit for Key-less Tables	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
Data-Change Audit BLOB columns	X				X			X	X	X	X	X	X	X	X	X	X		X
Real-time Email Alerts			X	X	X	X	X			X[1]	X[1]	X[2]	X[2]	X		X[3]	X[3]		X
Disable Triggers	X	X	X	X		X	X				X						X		
Automatic Data Audit Trail Purge	X	X	X	X		X	X				X	X	X	X		X	X		X
Automatic System Audit Trail Purge	X	X	X	X			X									X[5]	X[5]		X
System Audit Trail Archival	X	X	X	X			X		X	X	X					X[5]	X[5]		X
System Audit Trail Export	X	X	X	X			X		X	X	X					X[5]	X[5]	X	X
Data Audit Trail Archival	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
Data Audit Trail Export	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
Host Central Repository			X	X			X										X		X
Join Central Repository	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Database Security Management			X	X			X												
Effective Security Settings Analytics			X	X			X												
PCI/PII Data Search	X	X	X	X	X	X	X									X	X	X	X

1 – For ASE databases hosted on Windows NT platforms DB Audit uses *xp_sendmail* extended stored procedure. For ASE database hosted on UNIX platforms, DB Audit uses the host Operation System's mail command executed via the ASE *xp_cmdshell* extended stored procedure.

2 – Supported only with ASA running on Windows platforms.

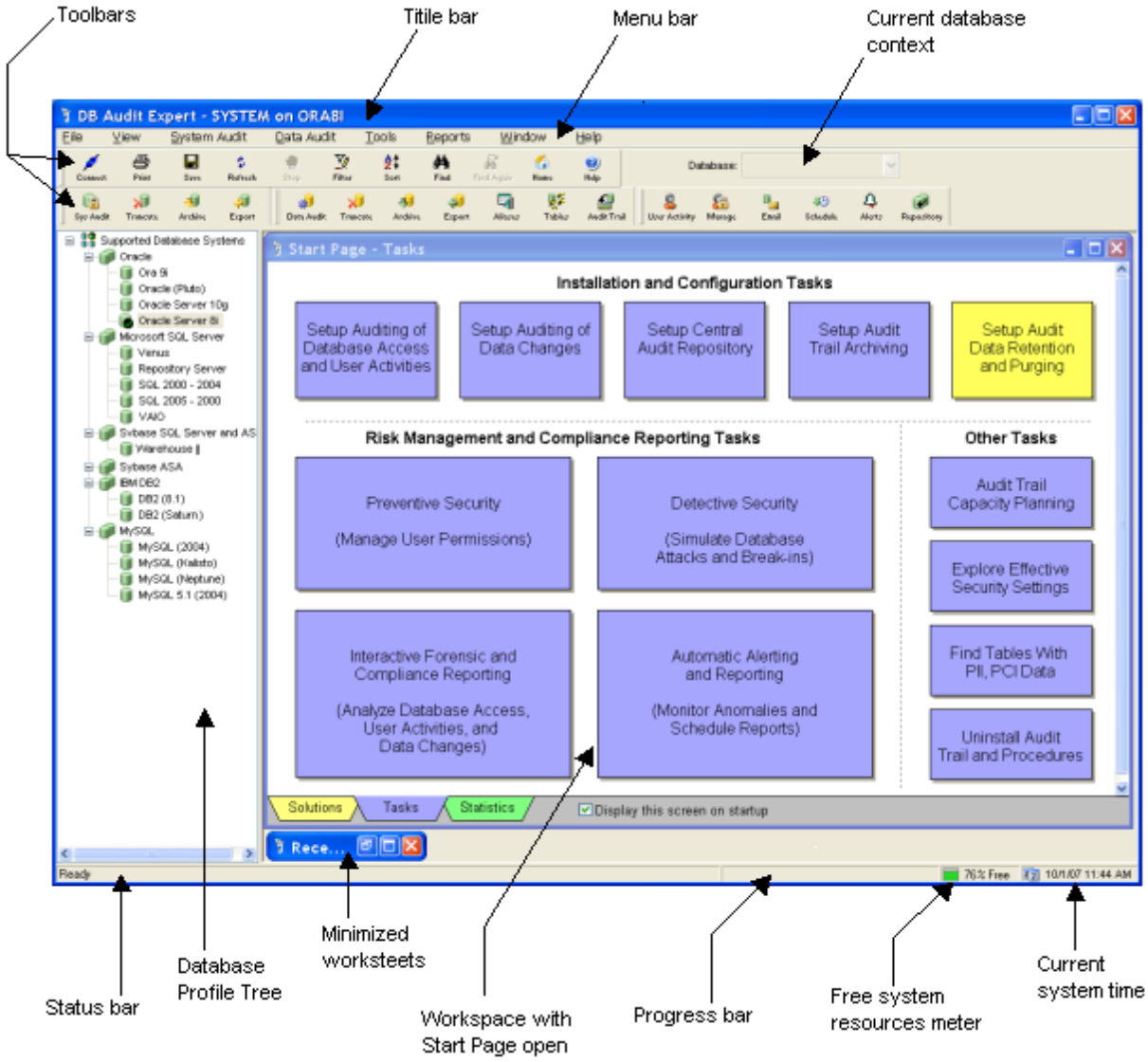
3 – Requires that the JavaMail package be installed on the system.

4 – Limited support.

5 – DB2 UDB for Linux, Unix and Windows only.

N/A – System or data-change auditing is not supported by DBMS

DB Audit Main GUI Controls



Toolbar

The toolbar displays buttons duplicating most frequently used commands in the menus. The buttons provide quick and easy access for those commands.

Active toolbar items appear as 2D graphical icons until you move the mouse pointer over them. Disabled menu items have a silhouetted, engraved appearance and do not respond to the mouse pointer.

Tip: To get help on the function of any toolbar button, rest the mouse pointer over the desired button for a moment. DB Audit Expert will display a tool-tip describing the function of the active button.

Title Bar

DB Audit Expert displays the login/user name and the server name for the active database connection.

Menu Bar

The Menu bar provides a standard Windows menu interface to DB Audit Expert. You can use shortcut keys to access certain menu commands quickly. The shortcut keys are displayed next to the menu item names (for example **Ctrl+F** for text search). All menu items also feature quick keyboard navigation keys. For example, pressing **Alt+F, C** displays the connection dialog. The navigation keys are highlighted by underline characters. In Windows the quick navigation keys are hidden until you use the Alt key to enable display of keyboard shortcuts (the underlined characters in menus and controls) and input focus indicators (the dotted rectangles around active objects).

Current Database Context

The **Database** drop-down box lists the databases available in the currently connected database server. Select the database you want to work with. When DB Audit Expert connects for the first time to a SQL Server or ASE server, it will detect the default database and select it in the drop-down box. On any subsequent sessions DB Audit automatically selects and switches to the most recently used database so you can resume working where you left off in the previous work session.

SQL Server and ASE:
The **Database** drop-down box is enabled only when you are connected to a database system supporting multiple databases such as Microsoft SQL Server, Sybase SQL Server and Sybase Adaptive Server Enterprise.

Database Profile Tree

The Database Profile Tree lists the supported database systems and defined database profiles for each system. When you specify a new profile name in the Connect To Database Dialog, DB Audit automatically adds the new profile to the tree. The profile information is saved between DB Audit sessions in the system registry. You can easily display or hide the list of supported database systems or profiles for a particular system.

- **To display or hide the list of supported database systems:**
  Double-click the Supported Database Systems item displayed in the top portion of the tree.

- **To display the list of database profiles defined for a system:**
  Click the plus sign (+) preceding the system name
  or
  Double-click the system name.

- **To hide the list of database profiles for a system:**
  Click the minus sign (-) preceding the system name
  or
  Double-click the system name.

- **To delete a profile:**
  Click the profile item in the tree and then choose the **File/Delete Profile** command from the menu. The profile as well as the related database connection and audit system configuration information is permanently deleted from the system registry.

Notes:
Deleting a database profile does not uninstall any associated DB Audit back-end objects from the database. Use **Tools/Uninstall Audit Repository** menu to uninstall DB Audit back-end objects. For more information, see the Back-end Uninstallation topic.

- **To Connect to the database specified in a particular profile:**
  Double-click the profile name. The Connect To Database Dialog will appear, enter your database password and then click the Connect button.

Status Bar

The status bar displays various context-sensitive messages as well as status messages that show the progress of lengthy operations.

Progress bar

The progress bar is displayed during lengthy DB Audit operations whenever a processing time estimate or amount of work estimate is available.

Workspace Area

The workspace area is where DB Audit displays audit reports and other worksheets. The workspace area is bounded by the Toolbars, Status Bar, Database Profile Tree and right edge of the DB Audit window. To resize the work area, resize the main DB Audit window.

Free System Resources Meter

This section displays a small resource meter reflecting the current amount of used and free system resources. The height of the green bar is proportional to the amount of free system resource. The bar turns yellow when less then 15% of resources are free and turns red when less then 10% of system resources are free. Monitor the free system resources meter when retrieving long audit reports. To prevent the system from becoming unstable, you should cancel the report retrieval before it consumes all free system resources.

Current Date/Time

This section displays current date and time.

DB Audit Start Page

The Start Page contains shortcuts to frequently used DB Audit features, reports and utilities. It also provides a one-page overview of supported DB Audit configurations as well as a one-page summary and details for current audit statistics.

Solutions

On the Solutions tab page you can review available components and decide which DB Audit configuration is most appropriate for your environment.

Tips:

- Click on the blue hyperlinks displayed on the top of the page to quickly jump to one of the available configurations
- Click on any element or component displayed on the configuration chart to display a context description of that component.

Tasks

The Tasks tab page provides shortcuts to frequently-used DB Audit features, reports and utilities. Click on any blue rectangle to activate the shortcut.

Important Note: Some shortcuts require an active database connection for DB Audit to know how to handle the shortcut properly and to know which function to launch. It is recommended that you connect to the database before using any shortcut. For information on how to setup database connections see CHAPTER 2, Connecting to Your Database

Statistics

The Statistics tab page provides a summary view of the audit trail. This page can be used to pull today statistics from any audited database and from a central repository server.

To get the statistics:

1. Fill in **Connect to Database Server** area. Enter the repository type, select one of the configured database connection profiles, and, if necessary, enter a user and password.
2. Click the **Retrieve Statistics** button to display summary statistics in the **Summary Statistics** area.
3. To see details for any given audit category, click on the hyperlinks in the **Summary Statistics** area. The detailed reports will appear on the bottom of the Statistics page.
4. To see statistics for a different server, select a different database profile and repeat steps 2 and 3.

CHAPTER 2: Connecting to Your Database

DB Audit Expert can connect to a database using either an ODBC interface or a native database driver. DB Audit Expert software includes native database drivers for Oracle, Sybase ASE and Microsoft SQL Server.

DB Audit currently does not include native drivers for ASA and DB2 connections so you should use ODBC connections to connect to these database systems. ODBC drivers for DB2 and ASA are available from IBM, Sybase and other vendors.

You can also use ODBC for Oracle and Microsoft SQL Server connectivity; however, the use of native drivers is recommended whenever available. Native drivers provide optimal performance as they do not use ODBC middleware.

Preparing your database for use

Preparing the database ensures that you will be able to access and use your data. The requirements differ for each database but in general, preparing a database involves the following steps:

1. If network software is required, make sure it is properly installed and configured at your site and on the client machine.

2. Make sure the required database server software is properly installed and configured.

3. Make sure the required database client software is properly installed and configured on the client workstation. (Typically, the client workstation is the one running the DB Audit front-end GUI.)

 **Important Note:** You must install the appropriate client software for your database server and operating system platform before you can connect to the database. See your database vendor for specific information on where to obtain and how to install the client software.

Installing the ODBC driver or native database driver

To connect the DB Audit to your database, you must install the appropriate ODBC driver, OLEDB driver or native database driver for your database. Select the desired driver or database interface when prompted to do so by the setup program.

Defining the ODBC data source

Data you access through an ODBC driver is referred to as an ODBC data source. An ODBC data source consists of the data and associated DBMS or file manager, an operating system and, if present, network software. You must define the ODBC data source to provide the driver with the information it needs for the database connection. (Defining an ODBC data source is also referred to as configuring the data source.) You can use the standard Windows ODBC Manager software to create and modify ODBC data sources.

To start the ODBC Manager, do the following:

From Windows Control Panel

1. Click the Windows **Start** button.
2. Select the **Settings** menu, then select **Control Panel**. The Control Panel window displays.
3. Double-click the **ODBC** icon.

Completing the ODBC setup dialog box

Define an ODBC data source by completing the ODBC Setup dialog box for the selected ODBC driver. The content and layout of the ODBC Setup dialog box varies for each driver, but most ODBC setup dialog boxes require you to supply the following information:

- Data source name and location
- Data source description (optional)
- Other DBMS-specific connection parameters

After you have created a data source, you can use it in the database profile you create in the Database Profile Tree. Refer to the DB Audit Main GUI Controls topic for details.

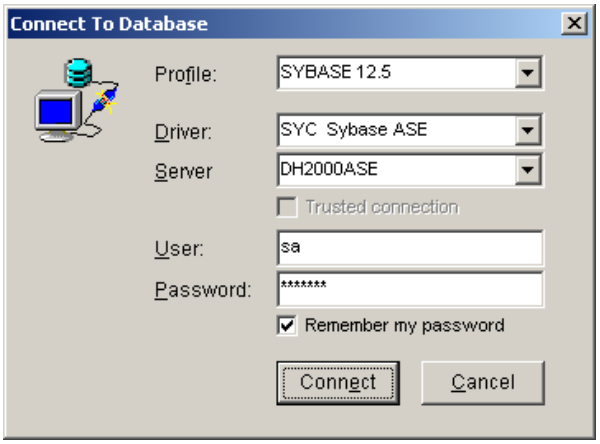
Troubleshooting the database connection

DB Audit supports two methods for tracing database connections in order to troubleshoot problems:

- **Database Trace** - The Database Trace tool records the internal commands that DB Audit executes while communicating with a database. Database Trace writes its output into a text file named PBTRACE.LOG, which is created in the Windows home directory. You can view the contents of the log file using any text editor. To enable database tracing, type "TRACE " (without quotes) in front of the chosen driver name on the Connect To Database Dialog.

- **ODBC Driver Manager Trace** - The ODBC Driver Manager Trace tool records information about the ODBC API calls made by DB Audit while connected to an ODBC data source. By default, the ODBC Driver Manager Trace writes its output either to a file named SQL.LOG, located in the Windows home directory, or to a log file you specify. You can view the ODBC Driver Manager Trace log at any time using any text editor.

Connect To Database Dialog



Use the Connect To Database dialog box to connect to a database server. Based on the information entered, DB Audit automatically creates a new database profile or updates an existing profile. The new or updated profile information is saved in the system registry.

A **database profile** is a named set of parameters stored in the system registry in the DB Audit key. The profile data contains both the connection parameters for a particular database system and the DB Audit configuration parameters for that system. Examples of DB Audit configuration parameters include the email server name, the name of the DB Audit repository database, and so on.

Choose the profile name and specify the required connection parameters:

Profile Name

Enter a descriptive name for a database you connect to, for example, “Data Warehouse,” or “Personal Dept. DB.” The profile name can contain any text up to 50 characters including spaces. If the profile already exists, select it from the **Profile** drop-down list and DB Audit will automatically populate the rest of the connection dialog fields.

Driver

Choose the appropriate driver name from the drop-down box. For ODBC connections choose ODBC.

Important Note: When choosing a native database driver for the connection, make sure you select the driver that matches the version of the database client software installed on your system, not the version of your database server software. For example, if you have Oracle SQL*Net 2 with Oracle 7.3 client files installed on your workstation and you connect to an Oracle 9i database, you should select the *O73 Oracle 7.3* driver for the connection.

Server

Choose the desired server name from the Server drop-down box. If you cannot find your server name in the list, type it in the edit field.

Trusted Connection

This option can be used only with Microsoft SQL Server connections. Check this option if your SQL Server security system is configured for Windows authentication.

Important Note: When Trusted Connection option is checked, both the User and Password fields

become disabled and cannot be changed. DB Audit will use your Windows network logon for the database connection

User

For non-trusted connections, provide the database user or login name for the server you are connecting to. For trusted connections, do not enter anything in this field.

Password

For non-trusted connections, provide the password for the user or login name. For trusted connections, do not enter anything in this field.

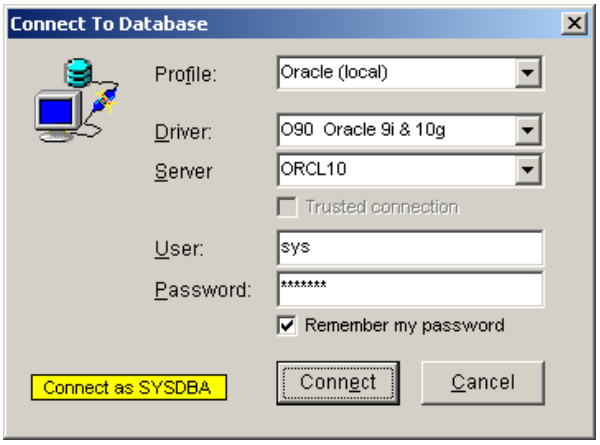
Remember my password

Use this option to allow DB Audit to remember your database password the next time you connect to the database. If this option is checked, the password is encrypted and stored in the local system registry along with other profile information.

Note: This information is only as secure as the secured access to your computer. If you share this computer with other non-privileged users, do not use this option as it compromises audit security.

Connect as SYSDBA

This option is only applicable to Oracle connections using Oracle 9i, 10g and 11g client software. DB Audit automatically connects you to the specified Oracle database as SYSDBA if you enter *SYS* for the user name and also select the **Oracle 9i, 10g or 11g** database driver. Specifying these parameters causes the *Connect as SYSDBA* highlighting to appear on the left side of the Connect To Database dialog as shown below.



You can still use the SYS user name for connections to older versions of Oracle client software; however these connections are always established as *Normal* connections, regardless of the selected driver.

CHAPTER 3: System Auditing

How it works

Depending on the brand and version of your DBMS, system-level audit functions provide the capability to audit various system events such as logins, logouts, server start/stop operations, remote procedure calls, accesses to database objects, and all actions performed by a specific user or role.

DB Audit Expert provides graphical point-and-click interfaces to DBMS native system-level audit functions. These interfaces rely heavily on the audit methods supported by the DBMS as well as on API and audit system implementation where applicable. Hence, system audit options supported by DB Audit are specific to each DBMS. Although the DB Audit graphical interfaces for different databases are similar, you should carefully study how your DBMS system-level audit is implemented so you are sure you understand what kind of changes need to be made and what the impact of those changes will be. Your DBMS Systems Administration Guide and System Security Guide are good starting points to understand how the system-level audit works, as well as how to install and enable the required DBMS support for system-level auditing.

Oracle: DB Audit Expert provides a rich graphical interface for Oracle system audit management and audit data analysis. All Oracle database systems support three standard types of auditing: SQL statement-level, privilege-level, and object-level auditing. Audit records can be written to the standard Oracle audit table SYS.AUD$ or to an external file. The three basic types of auditing can be performed by user, by successful or unsuccessful attempts, and by session or access time intervals. The standard system auditing does not include data-change auditing at the record or column level. DB Audit provides this missing functionality using advanced data-change audit methods and tools that work at the record and column levels. To learn more about data-change auditing in Oracle, refer to the CHAPTER 4: Data Change Auditing.

Through the use of the standard auditing capabilities, it is possible to audit virtually any system-event such as:

- User logon/logoff times and activity
- Execution of a specific SQL statement, such as auditing table manipulation operations
- Use of system privilege or privileges (there are over 100 privileges such as CREATE VIEW and SELECT ANY TABLE)
- Audit by object to see what was done to a specific object (tables, views, indexes, and so on)
- Audit a specific user or group of users
- Audit middle tier applications, such as access by another database
- Audit successful and/or unsuccessful events, including unsuccessful logons and object access

Notes:

- System-level database auditing is performed entirely on the server. DB Audit Expert uses Oracle DBMS SQL commands to access the system audit trail information and manage system audit configuration and settings.

You must use the audit_trail=DB parameter to enable auditing at any level. Because the audit_trail=DB parameter is not settable dynamically, you must first set the parameter in the instance parameters INIT.ORA file and then restart the instance. On UNIX systems this file is normally located in the dbs subdirectory under the ORACLE_HOME directory. On Windows systems, this file is named INITORCL.ORA and is normally located in the Database subdirectory under the ORACLE_HOME directory.

- You must have AUDIT ANY and CREATE TABLE privileges to manage system audit configuration and settings.

SQL Server: DB Audit Expert provides a rich graphical interface for SQL Server system audit management and audit data analysis.

DB Audit Expert uses the Microsoft SQL Server trace API to implement fully functional system auditing; however, system-level auditing is not supported in SQL Server versions 6.x and 7.0. This is because of changes made to the SQL Server trace API that are not backward compatible with these older versions.

All audit trail records are written to DB Audit's user-defined table, DB_AUDIT.SYS_AUDIT_TRAIL. System auditing does not include data-change auditing at the record or column level. DB Audit provides this functionality using advanced data-change audit methods and tools that work at the record and column levels. To learn more about data-change auditing in SQL Server refer to the CHAPTER 4: Data Change Auditing.

Through the use of the standard auditing capabilities, it is possible to audit virtually any system-event such as:

- Occurrence of any global, server-wide, security-related event
- Creating, deleting, and modifying database objects
- All actions by a particular user or all actions by users with a particular active role
- Granting or revoking database access
- Importing or exporting data
- Logins and logouts

Notes:

- System-level database auditing is performed entirely on the server. DB Audit Expert uses SQL Server extended procedures and T-SQL commands to access system audit trail information and to manage system audit configuration and settings. To enable auditing at any level, you must use DB Audit's graphical interface or web-based interface.
- You must have SA privileges in order to install, uninstall or manage system audit settings.

ASE: Starting with Version 10, Sybase supports system-level auditing in Sybase SQL Server and Adaptive Server Enterprise. However, different database server versions feature very different implementations of system audit functions. DB Audit Expert currently provides an interface to auditing functions for Sybase versions 11.0 and up. The DB Audit Expert GUI interface provides access only to the options available in the version of the Sybase server you are connected to. In addition to system-level auditing in ASE, DB Audit fully supports data-change auditing at the row and column levels. To learn more about data-change auditing in ASE refer to the CHAPTER 4: Data Change Auditing.

Through the use of the currently available auditing capabilities in Sybase 12.0 and later, it is possible to audit virtually any system-event such as:

- Global server-wide, security-relevant events
- Creating, deleting, and modifying database objects
- All actions by a particular user login or all actions by users with a particular role active
- Granting or revoking database access
- Importing or exporting data
- Logins and logouts

Notes:

- System-level database auditing is performed completely at the database back end. DB Audit Expert uses ASA SQL commands and system stored procedures to access system audit trail information and to manage system audit configuration and settings.
- You must have SSO_ROLE or SA_ROLE privileges to start and stop auditing, set up auditing options, or process audit data.

ASA: Starting with Version 7, Sybase supports system-level auditing in Sybase Adaptive Server Anywhere. DB Audit Expert currently does not provide a graphical interface to these functions. Support for ASA system-level auditing will be added in the next version.

To learn more about data-change auditing in Sybase Adaptive Server Anywhere, refer to the CHAPTER 4: Data Change Auditing.

DB2: Although IBM has been offering system-level auditing capabilities in DB2 for a long time, it does not provide a mainstream API that can be used to create graphical interfaces. The auditing utilities are available only by executing the db2audit command directly on the server host system. This command and its audit options are only available on **DB2 UDB for Linux, Unix and Windows**.

The DB2 system audit trail is not stored within the DB2 database and thus is not accessible using standard SQL query methods and not available for data mining and reporting. As a solution, DB Audit installs several stored procedures and repository tables that enable users to execute the db2audit command remotely on the server. The command is entered remotely by means of graphical interactive menus displayed in a DB Audit Management Console running on the client computer. The installed stored procedures also automate loading of audit trail data from binary audit log files into regular database tables, making it easy to report and analyze the system audit trail data.

To learn more about data-change auditing in DB2 refer to the CHAPTER 4: Data Change Auditing.

Guidelines

Although system auditing is relatively inexpensive, you should limit the number of audited events as much as possible in order to minimize the performance impact on the execution of audited statements and operations, as well as to minimize the size of the audit trail. Consider the following general rules when devising an auditing strategy:

1. Evaluate your purpose for auditing. Once you have a clear understanding of the reasons for auditing, you can devise an appropriate auditing strategy and avoid unnecessary audits. For example, suppose you have been asked to investigate suspicious database activity. Try to narrow the scope of the audit as much as possible by identifying the specific events to be monitored. What types of suspicious database activity do you suspect or have noticed? In a focused auditing strategy, you might choose to audit unauthorized deletions from or changes to arbitrary tables in the database. This strategy narrows the type of action being audited and the types of object affected by the suspicious activity.
2.
3. Audit knowledgeably. Audit the minimum number of statements, users, or objects required to get the targeted information. This prevents unnecessary audit information from obscuring more meaningful information and consuming valuable database storage space. Balance your need to gather sufficient security information with your ability to store and process it. For example, if you are auditing to gather information about database activity, determine exactly what types of activities you are tracking, audit only the activities of interest, and audit only for the amount of time necessary to gather the information you desire. Do not audit objects if you are only interested in each session's logical I/O information.
4. Because the volume of audit trail information usually grows very quickly, you should regularly archive audit records and purge the audit trail. Once you have collected the required information, archive the audit records of interest and purge the audit trail.

System audit trail management

Archiving system audit trail to a table

Select **System Audit/Archive To Table** command from the DB Audit Expert menu.

The Select Destination Table dialog appears. Either select the name of an existing archive table or enter the name of a new table. DB Audit Expert automatically creates the destination table if it does not exist. If you select an existing table, DB Audit Expert appends data from the system audit trail to the selected table. Note that if you have not truncated the system audit trail since the last archiving operation, you will now have archived the same audit trail records twice.

 **ASE:** If the audit system is configured to use multiple *sysaudit* tables, DB Audit will use SELECT…UNION ALL operations to archive data from all of them.

Exporting system audit trail to a file

Select **System Audit/Export to File** command from the DB Audit Expert menu.

The Select Export File dialog appears. Specify the name of the file to which you want to archive the system audit trail. If you select an existing file, DB Audit Expert will overwrite that file; otherwise, it will create a new file.

 **ASE:** If the audit system is configured to use multiple *sysaudit* tables, DB Audit will use SELECT…UNION ALL operations to export data from all of them.

Truncating system audit trail

Select **System Audit/Truncate** command from the DB Audit Expert menu.

 **ASE:** If the audit system is configured to use multiple *sysaudit* tables, DB Audit will truncate all of them.

Scheduling periodic audit trail purge

Depending on the number and type of events you are auditing, the volume of audit log data can grow very quickly in a short period of time. DB Audit supports scheduling of automatic data purge procedures to help you maintain reasonable limits on the size of audit trail tables. DB Audit purges all records from the audit trail tables that are time-stamped before a certain date and time.

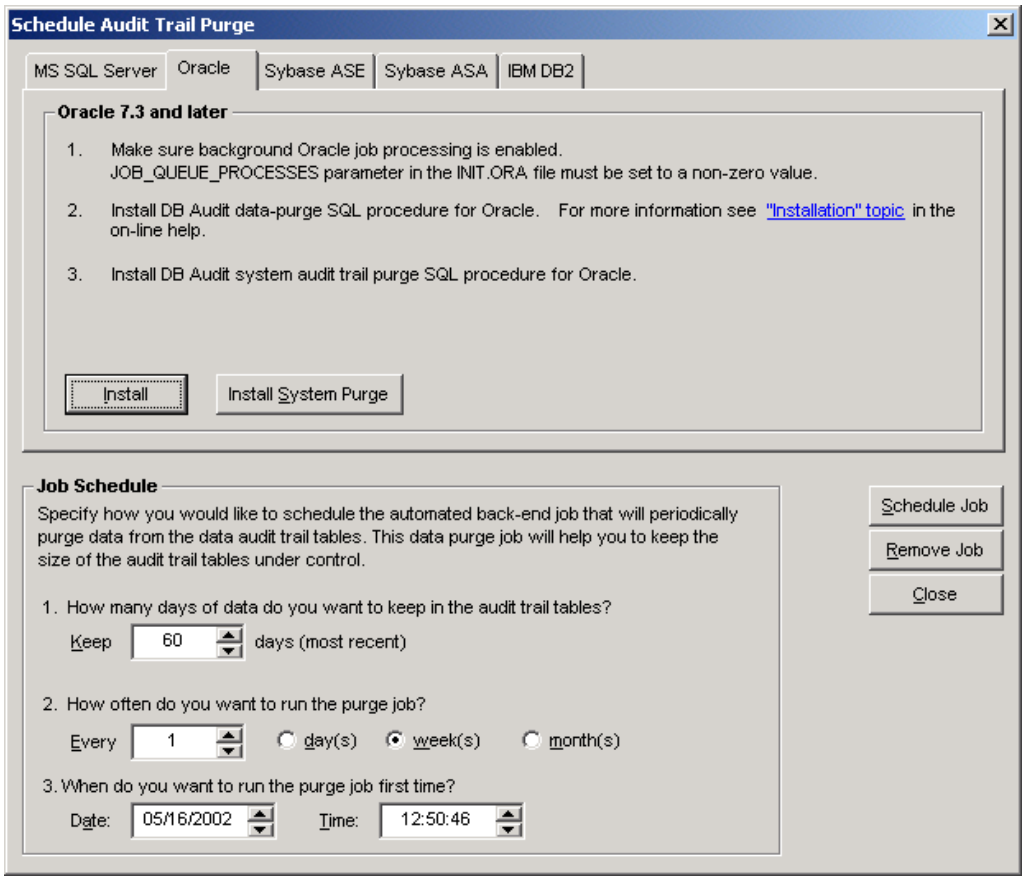
To install audit data purge procedure:

1. Click the **Schedule Periodic Purge** option on the **Tools** menu or click the **Schedule** button on the Toolbar.

2. Review the requirements for your database system as displayed on the screen. Make sure your database system has all the necessary components and settings for the data purge job.

3. Click the **Install** button to install the purge procedure.

    **Oracle:** Because Oracle does not support automated system audit log purging (SYS.AUD$ table), DB Audit provides an additional procedure to maintain the system audit log. If you wish, you can use the **Install System Purge** button to install the system purge procedure.



DB2: In some environments DB2 uses an external C compiler when compiling SQL stored procedures such as the data audit purge procedure, *DB_AUDIT.SP_AUDIT_SYSPURGE*. To install this procedure successfully, make sure your DB2 server settings are properly configured so that DB2 can locate and use the right compilers. It might be necessary to set the following environment variables:

```
DB2PATH=C:\SQLLIB
DB2_SQLROUTINE_KEEP_FILES=yes
DB2_SQLROUTINE_COMPILER_PATH=your compiler bin directory
```

In addition, you may need to set the correct compiler in the *\SQLLIB\function\routine\sqlproc.mak* file.

It is recommended that you close all connections to the database when editing this file. This is to ensure that DB2 does not have a lock on the file that would prevent saving changes. If you perform all the operations above and still get error messages while installing the data audit purge procedure, re-open *sqlproc.mak* and make sure the changes you made previously were saved. If necessary, repeat the procedure after making sure that there are no open connections to the DB2 database. Also make sure this file's attribute is not set to read only at the end of the edit.

To schedule the purge job:

Oracle, SQL Server: Because of the native database support for job scheduling available in

Oracle and in Microsoft SQL Server, you can use the DB Audit GUI to setup the purge job.

1. Specify parameters for the purge job as instructed on the screen.

2. Click the **Schedule Job** button to create a new job or click the **Remove Job** button to remove an existing data purge job.

ASE, ASA, DB2: Most versions of ASE, ASA and DB2 do not support scheduling using database server facilities. You must use either a standalone scheduling utility such as 24x7 Scheduler, or available host operating system scheduling functions to setup periodic runs of the DB Audit data purge procedure.

For example, on Windows NT systems you can use the *AT* command. On UNIX systems you can use *crontab* to schedule unattended run of a batch job using the *ISQL* utility. Consult your database documentation for details on how to use *ISQL* utility in a batch mode and your server operation system documentation for details on batch job scheduling.

The name of the data audit purge procedure is *DB_AUDIT.SP_AUDIT_SYSPURGE*.

Note: In ASE databases this procedure can only be installed and run if the system supports dynamic SQL executed from within stored procedures (ASE 12.0 and later).

Tip: To schedule *DB_AUDIT.SP_AUDIT_SYSPURGE* stored procedure runs using 24x7 Scheduler, you can either create a **program job** to run *ISQL* utility or a **database job** that directly connects to the database and executes the appropriate SQL command.

Examples

1. To run *DB_AUDIT.SP_AUDIT_PURGE* procedure in ASE using a **program job**:

   Create a new job and select the “program job” option. For the job command line, enter the following:

   ```
   isql [-S server] [-U user] [-P password] [-D repository] [-i inputfile]
   ```

   In this command, replace *'server', 'user',* and *'password'* parameters with the Sybase ASE server name, user name and password you use to connect to the database. Replace *'repository'* parameter with the name of the database you chose for the DB Audit repository tables. Replace the *'inputfile'* parameter with name of an existing text file. The input file must contain the following two lines of text:

   ```
   exec db_audit.sp_audit_purge @days = [n]
   go
   ```

   Replace [n] with the number of days of audit history you want to keep in the database. Select the desired job schedule and save changes.

2. To run *DB_AUDIT.SP_AUDIT_PURGE* procedure in ASE using a direct **database job**:

   Create a new database profile using the **Database Profiles** option on the **Tools** menu. Specify the required connection parameters and profile name (for example, "Sybase ASE"). Create a new job and select "database job" option. For the job SQL, enter the following:

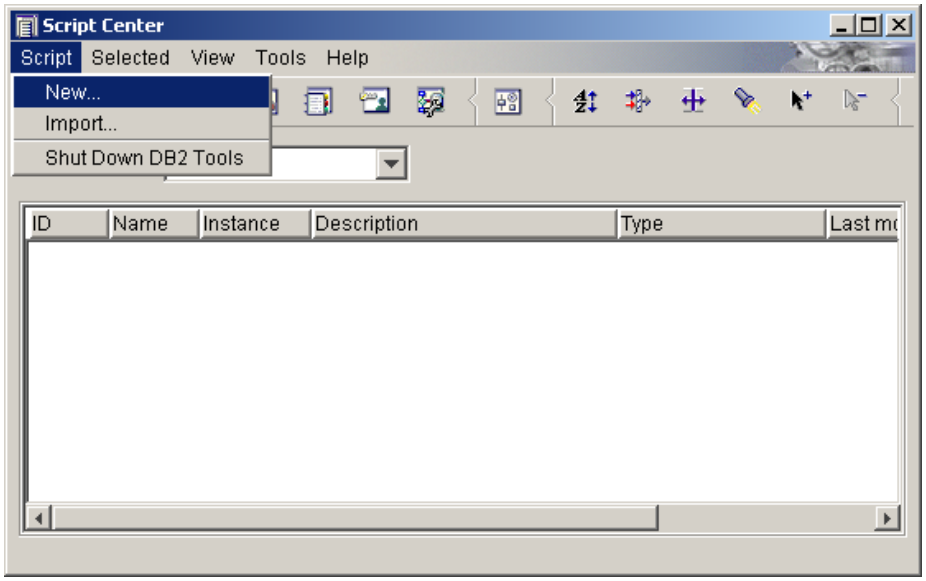
   ```
   execute db_audit.sp_audit_purge @days = [n]
   ```

   Replace [n] with the number of days of audit history you want to keep in the database. Choose the previously created database profile "Sybase ASE". Select the desired job schedule and save

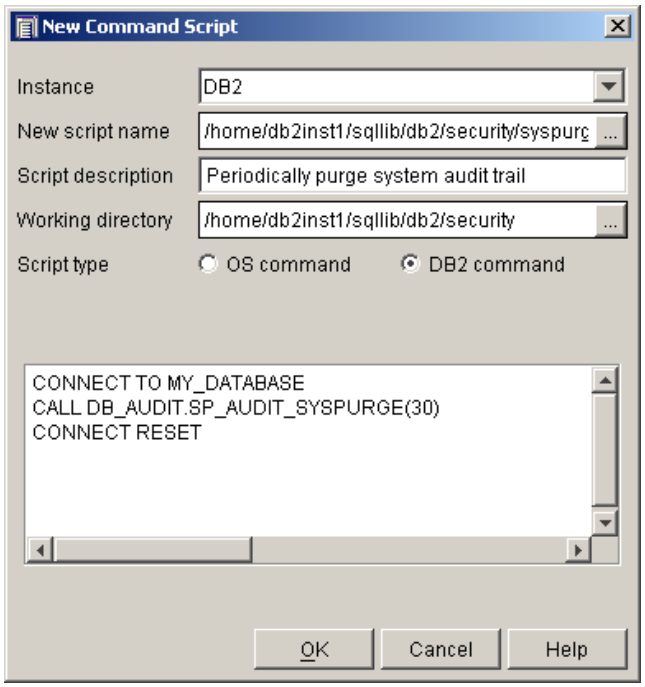
your changes.

DB2: Some versions of DB2 support job scheduling the use of built-in database server facilities. In these versions, jobs can be scheduled using the DB2 Script Center or DB2 Task Center tools. The following example demonstrates how to schedule a data audit trail purge procedure using DB2 Script Center.

1. Start DB Script Center. Click the **Script/New** menu to create a new job.



2. When the **New Command Script** dialog appears, enter the required job properties and a short descriptive phrase for the script. You can choose any location for the script file. Following is an example of a **New Command Script** dialog with valid values..



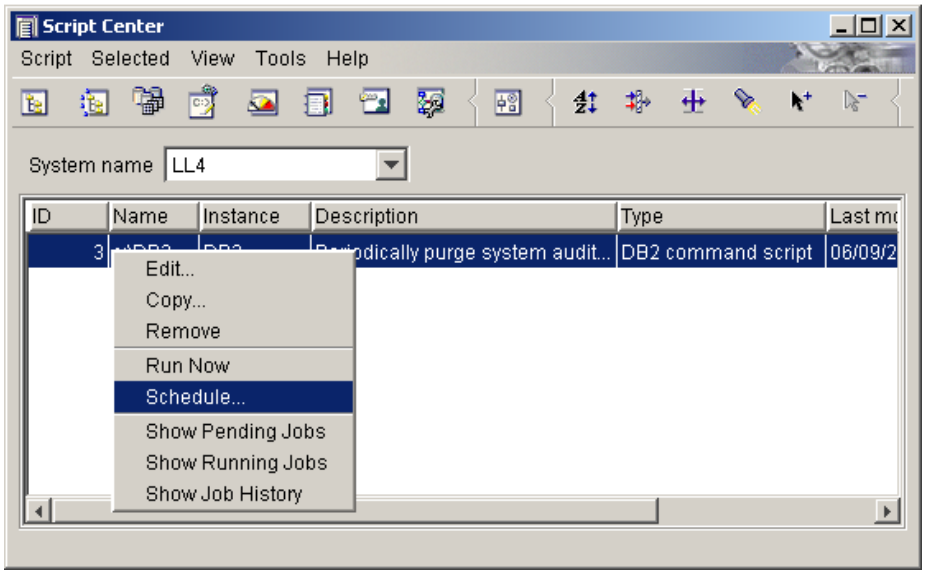
For the command script you must enter the following script

```
CONNECT TO MY_DATABASE
CALL DB_AUDIT.SP_AUDIT_PURGE(30)
CONNECT RESET
```

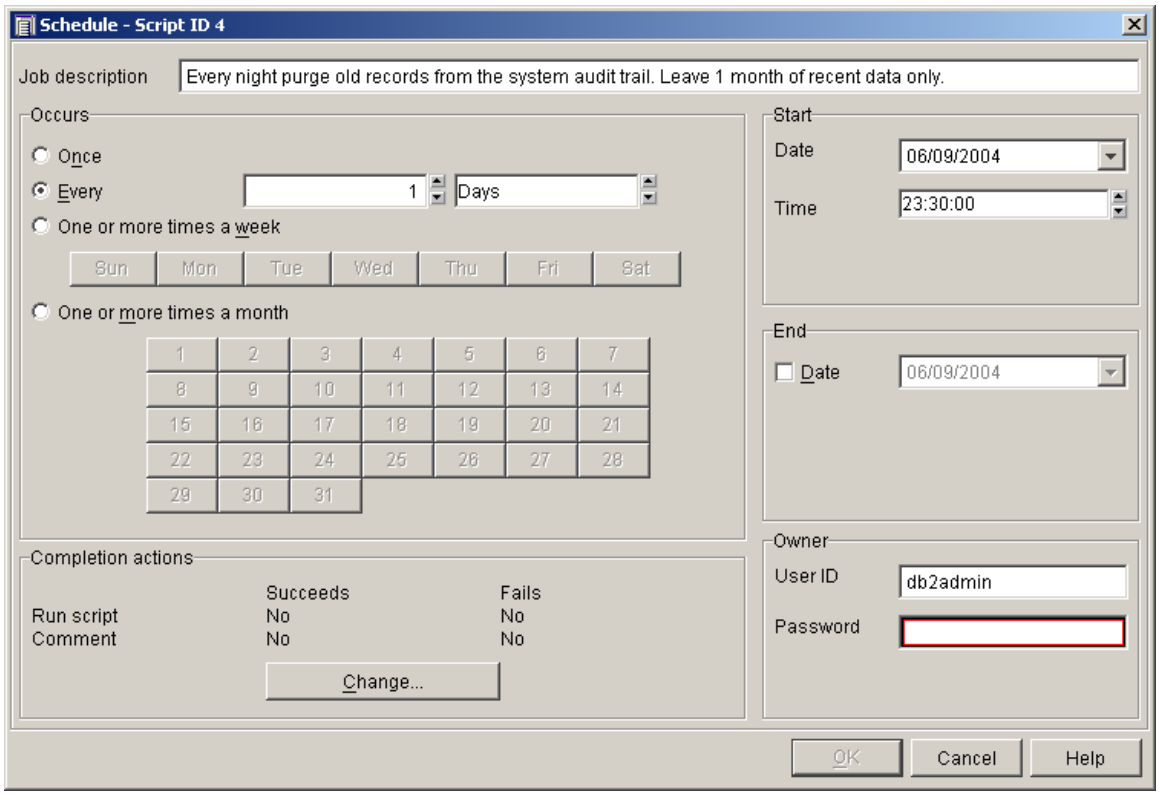
Note: In this script you must replace **MY_DATABASE** with the actual name of your DB2 database. Replace 30 with the number of days of audit history that you want to keep in the database.

Click the **OK** button to close the script dialog.

3. Now you need to schedule the created script. Right-click on the job line and select **Schedule** from the popup menu.



The **Schedule Script** dialog displays.



a) Enter descriptive schedule name; for example, *"Every night purge old records from the system audit trail. Leave 1 month of recent data only."*

b) Enter schedule occurrence. Recommended setup is to run this job once every day during "quiet" database hours.

c) Enter the job's initial start date and time.

d) Enter job owner information. This is the name and password of the user account that will be used to run the job. The account must have sufficient permissions to delete data from DB Audit repository tables.

e) Click the **OK** button to close the Schedule dialog.

That's it. The job is now scheduled and you can close the Script Center.

Scheduling periodic audit trail archiving to files

DB Audit supports automatic archiving of audit trail data to operating system files. Archiving can help you:

- Reduce storage costs by moving audit trail data to cheaper storage medium
- Increase performance and throughput of audit reports
- Reduce database operation costs by providing faster backup and restore times.

Tip: Archiving can be also used for consolidating audit trails from multiple database servers to one central location. The archived files can be transferred to a central server and loaded into a centralized audit database system, located in a secure area accessible only to authorized personnel.

Archiving procedures are designed for high performance processing. They support selective and incremental archiving of audit trail data. By default, archiving procedures pick up audit records that have been added and time-stamped since last archival process. They also support integrated purging of archived data after successful completion of each archive operation.

Archiving may be setup as a scheduled job that runs automatically at a specified time.

Note: If you choose to purge audit trail data during archiving, make sure you do not schedule a separate purging procedure described in the previous topic Scheduling periodic audit-trail purge

To install audit data archiving procedures:

1. Click the **Tools/Schedule Periodic Archiving** menu.

2. Review the requirements for your database system as displayed on the screen. Make sure your database system has all the necessary components and settings for the data archival job.

3. Click the **Install** button to install the necessary procedures. The following table and three stored procedures will be installed in the selected audit repository database:

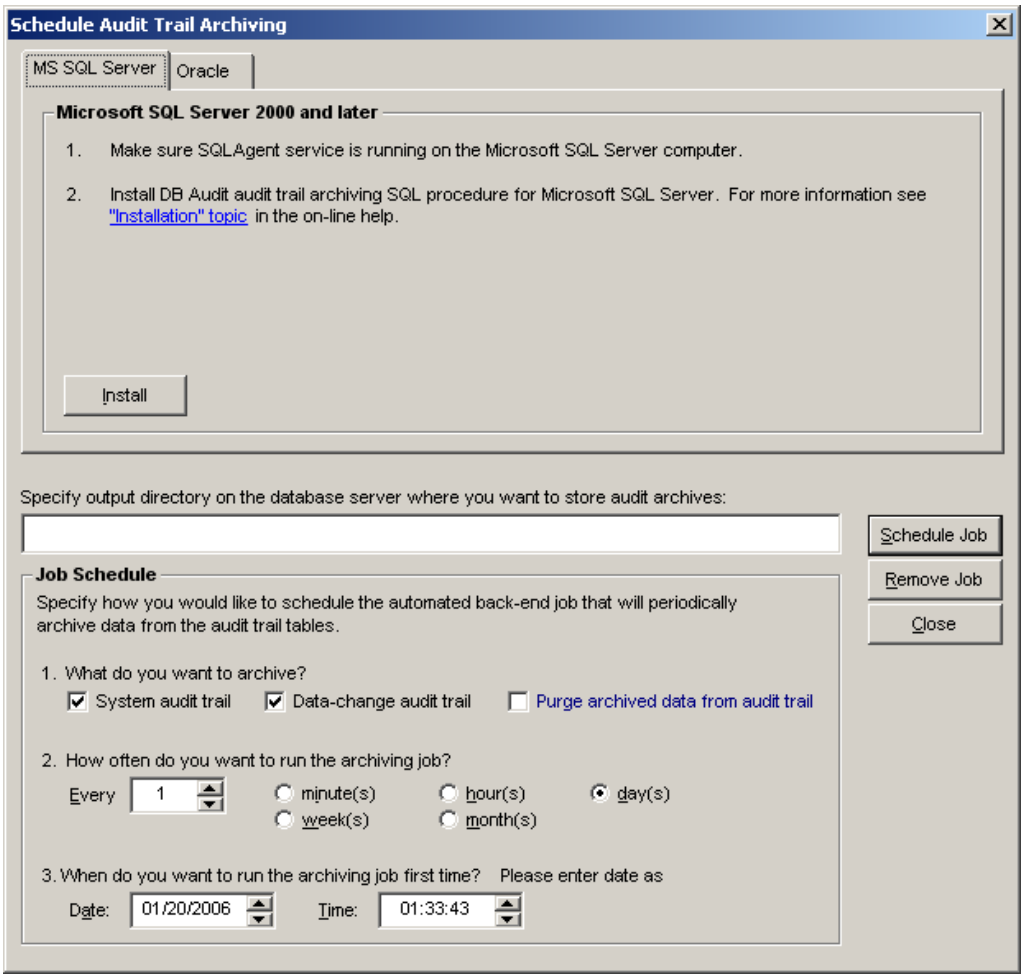
4. -- Table DB_AUDIT.ARCHIVE_TIME—Used to keep track of the last archiving event

   -- Stored procedure DB_AUDIT.SP_LOG_MESSAGE—Used to write progress of work messages to the processing log file maintained in the user-selected directory

   -- Stored procedure DB_AUDIT. SP_DUMP_TO_LOG—Used to write data from the given audit trail table to an operating system file. The file name is calculated using supplied procedure parameters.

   -- Stored procedure DB_AUDIT. SP_AUDIT_ARCHIVING—The main entry point procedure that internally calls DB_AUDIT.SP_LOG_MESSAGE and DB_AUDIT.SP_LOG_MESSAGE. This procedure must be scheduled.

   DB_AUDIT. SP_AUDIT_ARCHIVING has four parameters, all of which can be entering using the DB Audit Management Console interface. The following screen shot shows how to enter these parameters:



To schedule the archival job:

Oracle: Because of the native database support for job scheduling available in Oracle, you can use the DB Audit GUI to set up the audit data archival job:

1. Specify parameters for the archival job as instructed on the screen.

2. Click the **Schedule Job** button to create a new job, or click the **Remove Job** button to remove an existing job.

Note: The audit trail archiving procedure utilizes the **xp_cmdshell** system procedure to run the BCP process for fast exporting of audit data to operating system files. If you continue, this entire process will be scheduled and run using the SQL Agent. Be aware that the archiving procedure must be run using SA or a similar administrative account. If you attempt to run the SQL Agent under a low-privileged account, the audit trail archiving will fail. It is recommended that you use the Alert Center to securely schedule and run the archiving procedure.

SQL Server: Because of the native database support for job scheduling available in Microsoft SQL Server, you can use the DB Audit GUI to setup the audit data archival job. However, in certain situations, you may want to use another scheduling utility such as the Alert Center utility, which is available in the Enterprise version of DB Audit, or 24x7 Scheduler software, which can be obtained separately.

To schedule an audit trail archival job using SQL Agent:

1. Specify parameters for the archival job as instructed on the screen.

2. Click the **Schedule Job** button to create a new job or click the **Remove Job** button to remove an existing job.

Archiving audit trail to centralized audit repository

DB Audit supports automatic archiving of local audit trail data to central repository system residing on a different database server.

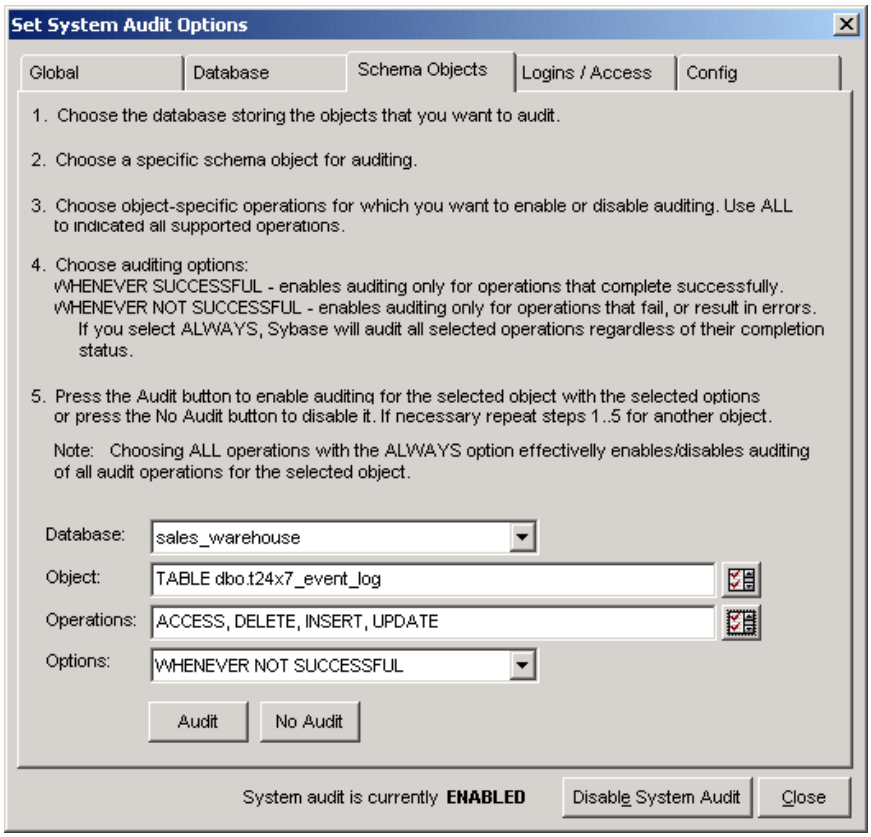
See CHAPTER 8: Central Audit Repository for information on how to configure the central audit repository system and how to setup the Alert Center to periodically archive local audit-trail to the central repository and then truncate the local audit-trail.

Oracle: Configuring System Audit Options

Depending on the auditing options set, audit records can contain different types of information but all auditing options generate the following information:

- The user who executed the audited statement
- The action type which indicates the audited statement executed by the user
- The object or objects referenced in the audited statement
- The date and time when the audited statement was executed

To select databases, operations, SQL statements, objects, or users for auditing, use the **System Audit/Set Audit Options** command from the DB Audit Expert menu. Follow the instructions provided by the **Set System Audit Options** wizard. The **Set System Audit Options** wizard is a graphical interface that helps you quickly set up your audit options and get up to speed quickly. Different database systems support different options. The wizard displays only those options available for the database system you are currently connected to.



For more information on the system audit trail and system audit options supported by your DBMS, refer to your DBMS documentation.

Enabling system audit

To manage audit options for Oracle, you must connect to your database as a user with AUDIT SYSTEM privileges. Use SYS or a similar logon with sufficient privileges to ensure that the audit setup process is error-free.

Important Note: Any authorized database user can set statement, privilege, object and session auditing options at any time, but Oracle does not generate audit information for the database audit trail unless database auditing is enabled. The AUDIT and NO AUDIT buttons only enable/disable selected auditing options; they do not enable/disable auditing as a whole.

To turn auditing on and control whether Oracle generates audit records based on the audit options currently set, set the initialization parameter AUDIT_TRAIL.

To enable DB Audit compatible auditing, you must set the AUDIT_TRAIL=DB parameter in the Oracle instance parameters INIT.ORA file and then restart the instance. This is necessary because the AUDIT_TRAIL=DB parameter is not settable dynamically. On UNIX systems this file is normally located in the dbs subdirectory under the ORACLE_HOME directory. On Windows systems, this file is named INITORCL.ORA and is normally located in the Database subdirectory under the ORACLE_HOME directory.

Disabling system audit

To disable system auditing, you must set the AUDIT_TRAIL=FALSE parameter in the Oracle instance parameters INIT.ORA file and then restart the instance. This parameter is not settable dynamically.

Setting system audit options

To launch the System Audit Configuration screen, click the **System Audit/Set Audit Options** command from the DB Audit Expert menu. The DB Audit GUI presents the Oracle auditing options organized into three groups:

1. **SQL Statements** – Options in this group are used to track the occurrence of SQL statements and privilege usage in subsequent user sessions. You can track the occurrence of a specific SQL statement or of all SQL statements authorized by a particular system privilege. Auditing operations on SQL statements apply only to subsequent sessions, not to current sessions. Both successful and unsuccessful operations can be tracked.

2. **Schema Objects** – Options in this group are used to track operations on a specific schema object. Auditing operations on schema objects apply to current sessions as well as to subsequent sessions. Both successful and unsuccessful operations can be tracked.

3. **Sessions** – Options in this group are used to track the occurrence of database logons and logoffs. Both successful and unsuccessful logons can be tracked.

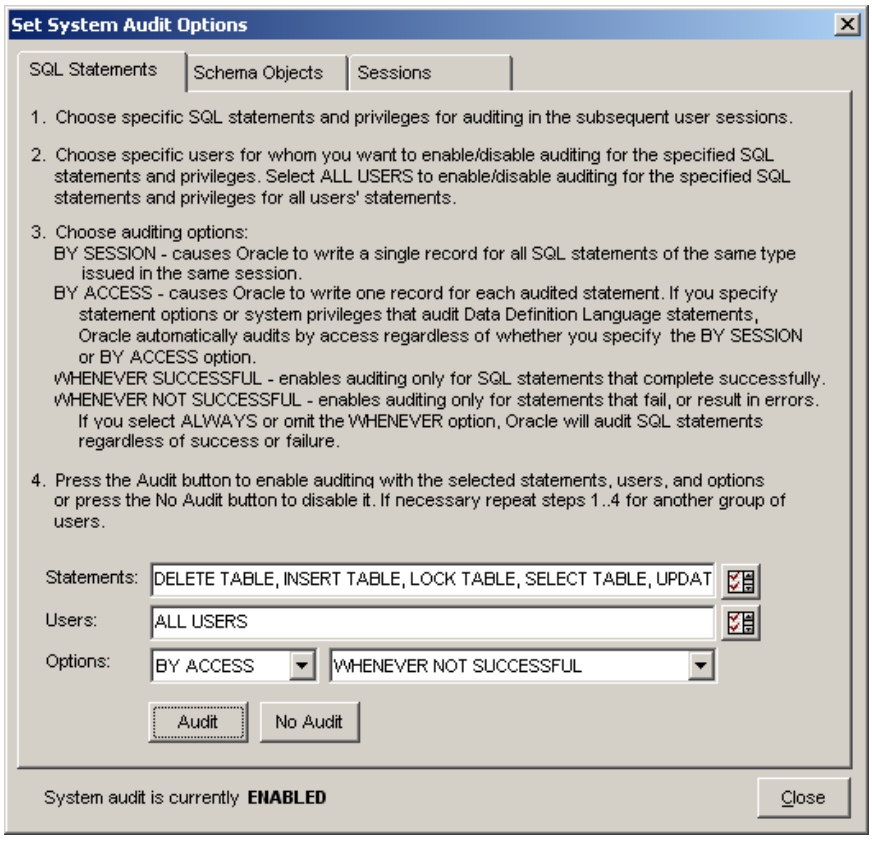
Each group of auditing options is displayed on a separate tab page. DB Audit provides context help and usage instructions directly on the auditing options screen.

 **Tips:**

- You can use **Enabled System Audits** reports from the **Reports** menu to find out which system-level auditing options are currently enabled in the database.
- Database Java schema objects (sources, classes, and resources) are considered the same as procedures, functions, and packages for the purpose of setting audit options.

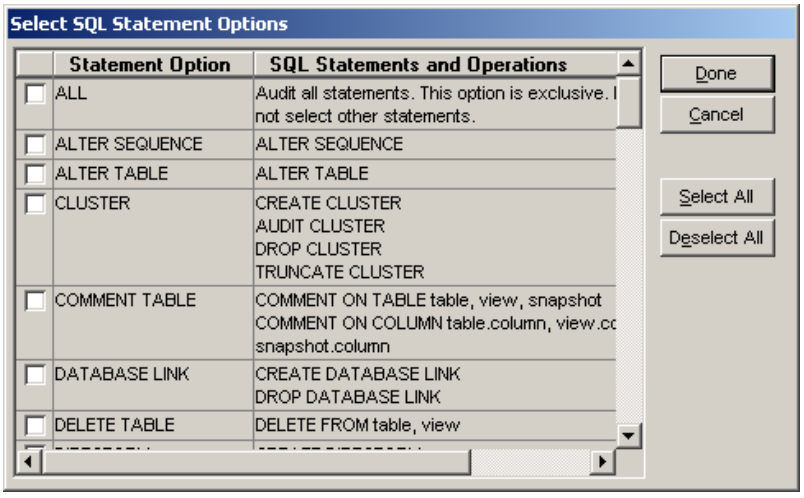
Setting SQL Activity Audit

The following screenshot shows typical selections for auditing SQL statements and privileges.



To setup SQL Activity auditing:

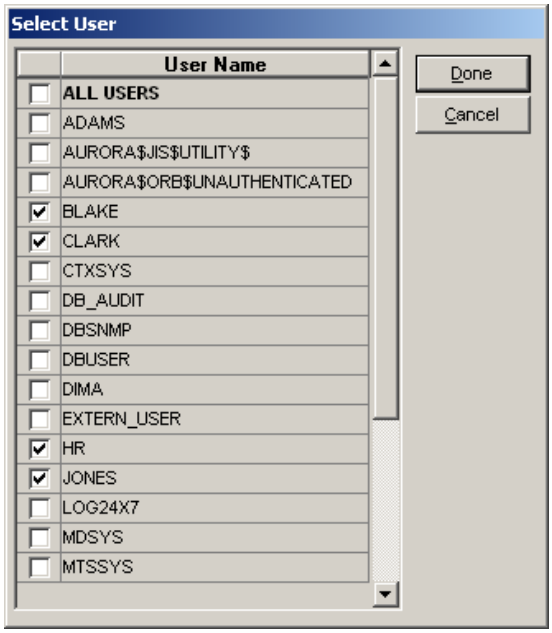
1. Activate the **SQL Statement** tab page. Choose specific SQL statements and privileges for auditing in the subsequent user sessions. To open the **Select SQL Statement Options** dialog, click the **Lookup** button  displayed on the **Statements** line.



To select one or more statement options, click the appropriate checkboxes in the left-most

column. To quickly select all listed options, click the **Select All** button. To quickly deselect all previously selected options, click the **Deselect All** button. Click the **OK** button to return to the previous screen with all selected options. To cancel the dialog without making any changes click the **Cancel** button.

2. Choose the users for whom you want to enable or disable auditing for the specified SQL statements and privileges. Click the **Lookup** button  displayed on the **Users** line to open the **Select Users** dialog. To change the setting for one or more users, click the appropriate checkboxes in the first column. To select all users, click the **All Users** checkbox at the top of the list.



3. Choose auditing methods and options using the two drop-down lists displayed at the bottom of the screen.

   Choose the BY SESSION audit method if you want Oracle to write a single record for all SQL statements of the same type issued in the same database session. Choose the BY ACCESS audit method if you want Oracle to write one record for each audited statement. If you specify statement options or system privileges that audit Data Definition Language statements, Oracle automatically audits by access regardless of whether you specify the BY SESSION or BY ACCESS method.

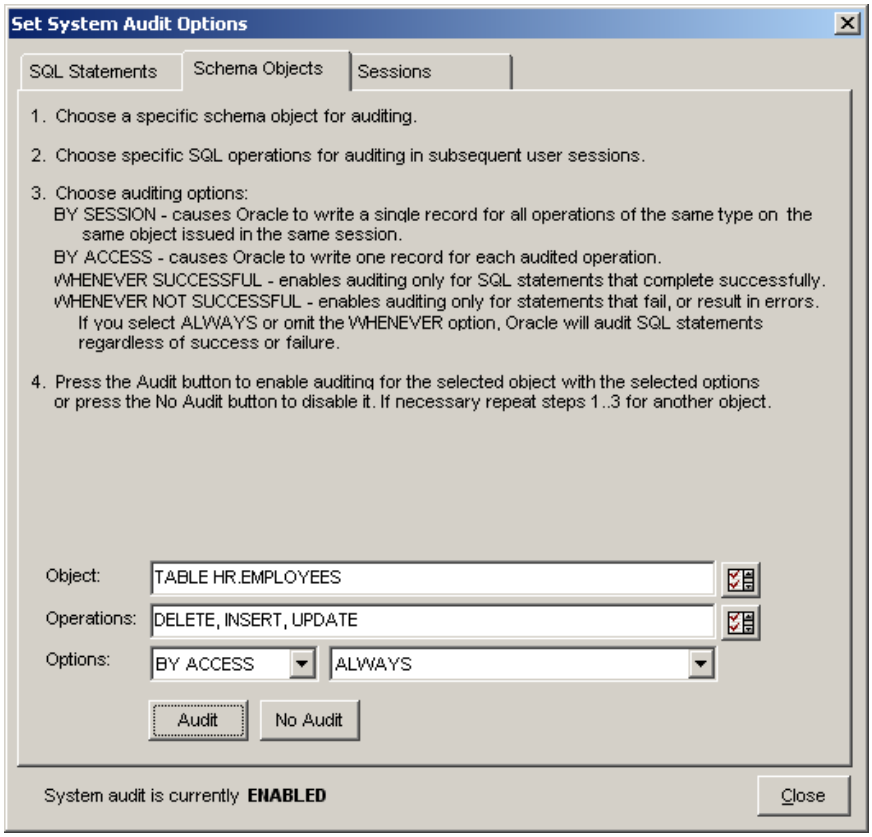
   Choose the WHENEVER SUCCESSFUL option to enable auditing only for SQL statements that complete successfully. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for statements that fail or that generate errors. If you select ALWAYS or leave the **Options** field blank, Oracle audits SQL statements regardless of success or failure.

4. Press the **Audit** button to enable auditing with the selected statements, users, and options or press the **No Audit** button to disable it. If necessary, repeat steps 1 through 4 for another group of users.

   Tip: You can select different audit options for different users and user groups. For example, you can audit all table SELECT queries for user JOHN and only failed LOCK TABLE queries for user SCOTT.

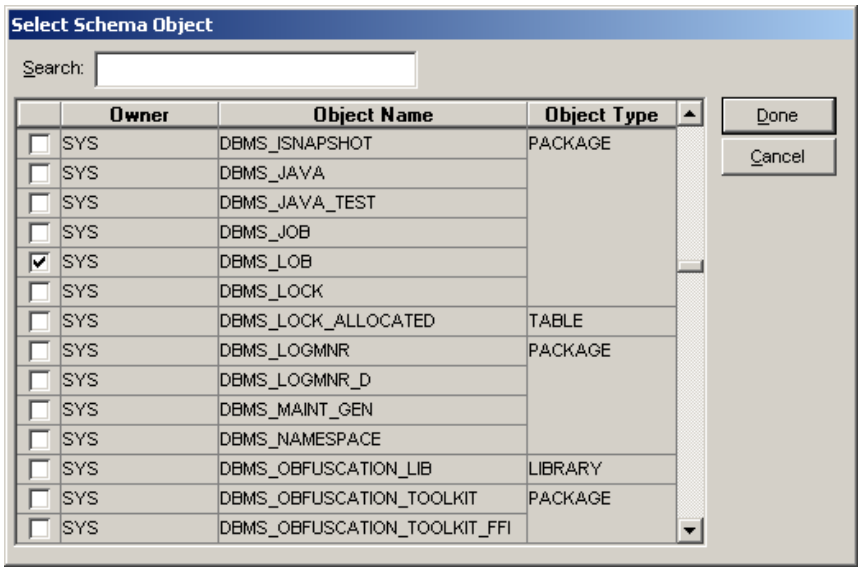
Setting Schema Object Audit

The following screenshot demonstrates typical selections for auditing access to Schema Objects.



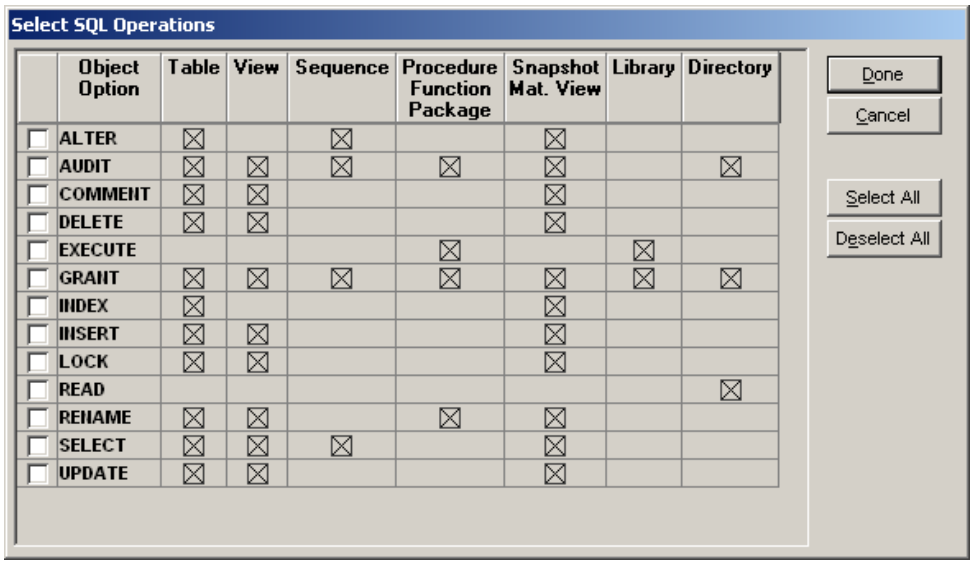
To setup schema object auditing:

1. Activate the **Schema Object** tab page. Choose a specific schema object for auditing. Click the **Lookup** button  displayed on the **Object** line to open the **Select Schema Object** dialog.



Select the object to be audited by clicking the checkbox in the first column, then click the **OK** button to return to the previous screen. To cancel the dialog without making any changes, click the **Cancel** button.

2. Choose specific SQL operations for auditing in subsequent user sessions. Click the **Lookup** button  displayed on the **Operations** line to open the Select SQL Operations dialog. The Select SQL Operations dialog lists all supported database operations applicable to each object type. Select one or more applicable options for the object selected in step 1. Note that the ☒ symbol indicates applicable options for each object type.



Click the **OK** button to return to the previous screen with the selected options.

3. Choose auditing methods and options using the two drop-down lists at the bottom of the screen.

   Choose the BY SESSION audit method if you want Oracle to write a single record for all SQL statements of the same type issued in the same database session. Choose the BY ACCESS audit method if you want Oracle to write one record for each audited statement. If you specify

statement options or system privileges that audit Data Definition Language statements, Oracle automatically audits by access regardless of whether you specify the BY SESSION or BY ACCESS method.

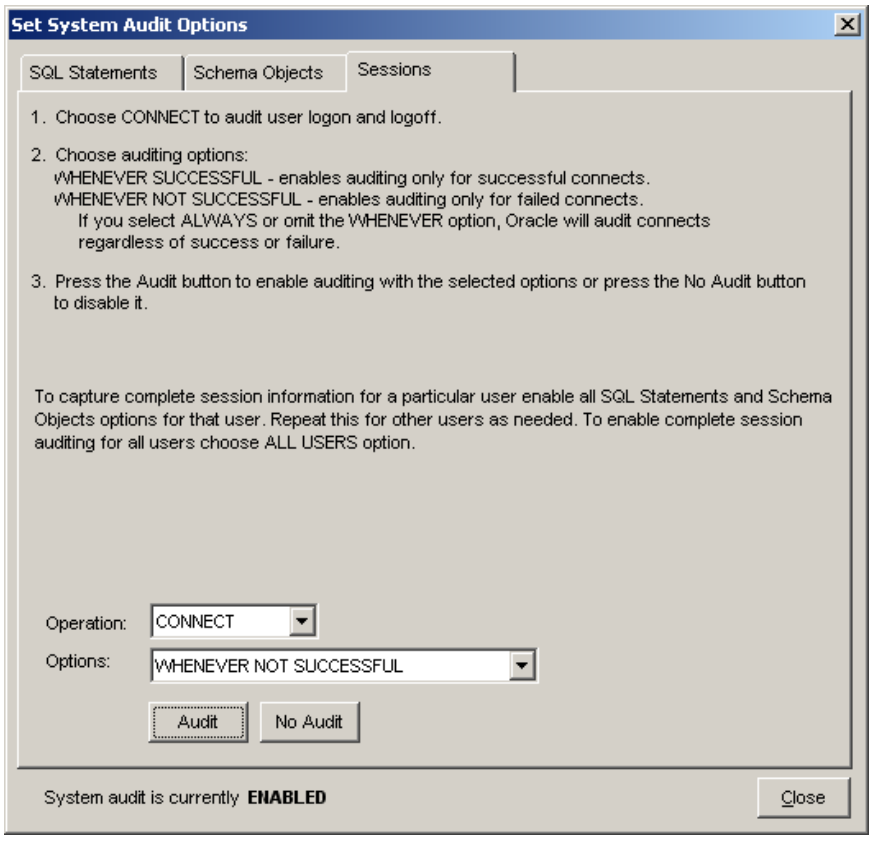
Choose the WHENEVER SUCCESSFUL option to enable auditing only for SQL statements that complete successfully. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for statements that fail or that generate errors. If you select ALWAYS or leave the **Options** field blank, Oracle audits SQL statements regardless of success or failure.

4. Press the **Audit** button to enable auditing for the selected object or press the **No Audit** button to disable auditing. If necessary, repeat steps 1 through 4 for another schema object.

Tip: You can select different audit options for different schema objects.

Setting Session Audit

The following screenshot demonstrates typical selections for Session auditing.



To setup session auditing:

1. Activate the **Session** tab page. Choose CONNECT as the type of the operation to audit (CONNECT is the only operation supported for session auditing).

2. Choose auditing options using the **Options** drop-down list at the bottom of the screen.

Choose the WHENEVER SUCCESSFUL option to enable auditing only for successful database connections. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for failed connections. If you select ALWAYS or leave the **Options** field blank, Oracle audits all database connections regardless of success or failure.

3. Press the **Audit** button to enable auditing with the selected options or press the **No Audit** button to disable auditing.

Configuring Advanced Options for Oracle

 **Oracle:** This section describes advanced auditing options available in Oracle databases only. To access and configure these options, start the DB Audit GUI console, then click the **System Audit/Advanced Options** menu.

The advanced options are described below.

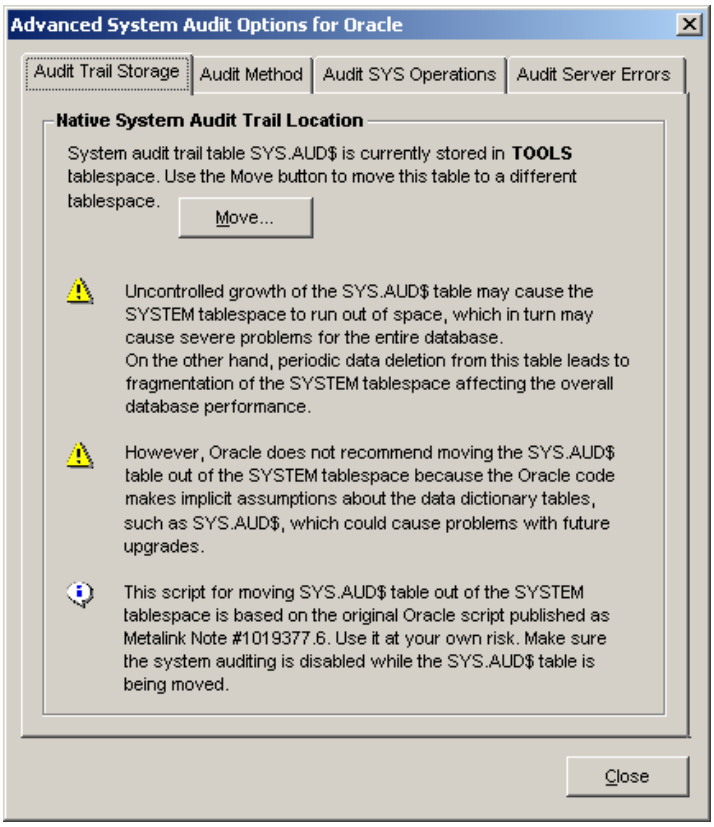
Moving Oracle native system audit trail table out of the SYSTEM tablespace.

The amount of output generated by the Oracle system audit can be quite large. All system audit data is stored in a single table SYS.AUD$, that is stored in the SYSTEM tablespace along with other Oracle system tables. Allowing the uncontrolled growth of the SYS.AUD$ table could cause the SYSTEM tablespace to run out of space, which in turn could cause severe problems for the entire database. For example, the database may hang if all free space in the SYSTEM tablespace were to be used up by audit records. This is why it is important to move the SYS.AUD$ audit table from the SYSTEM tablespace.

Periodically deleting old data from the SYS.AUD$ table is not a good solution for the space problem because it leads to fragmentation of the SYSTEM tablespace which, in turn, may affect overall database performance.

In its system documentation, Oracle is somewhat unclear about whether the SYS.AUD$ table should be moved out of the SYSTEM tablespace. In the System Administrators manual, it says explicitly that Oracle does not recommend moving the SYS.AUD$ table out of the SYSTEM tablespace. This is because Oracle code makes implicit assumptions about the data dictionary tables, such as SYS.AUD$, which could cause problems with future upgrades. However in published Oracle DBA courses and technical support bulletins, it recommends moving the table and provides instructions on how to do that.

DB Audit provides a graphical interface for moving the SYS.AUD$ table with just a few mouse clicks. If necessary, the table can be easily moved back to the SYSTEM tablespace at a later time.



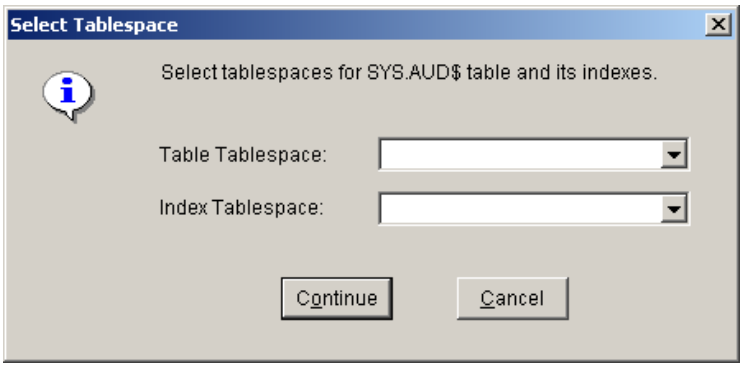
Use the steps listed below to move the SYS.AUD$ table to a different tablespace.

Note that it is important that you perform the move operation either when the auditing is turned off or during database maintenance windows when database usage is low.

1. Start DB Audit GUI console and connect to the database.

2. Click the **System Audit** menu, then click **Advanced Options** menu. The **Advanced Options** dialog is displayed.

3. Click the **Move** button. The Select Tablespace dialog is displayed.



4. Choose non-SYSTEM tablespace for the SYS.AUD$ table and its indexes, then click the **Continue** button.

Tip: In Oracle 8.0 and later, DB Audit uses simple ALTER TABLE and ALTER INDEX commands to move the table and indexes. In Oracle 7.x, it uses the following four-step procedure to move the table:

(1) Create a new temporary AUD$_NEW table

(2) Transfer data from the system audit trail table into the temporary AUD$_NEW table

(3) Rename the system SYS.AUD$ table to AUD$_OLD

(4) Rename AUD$_NEW to AUD$

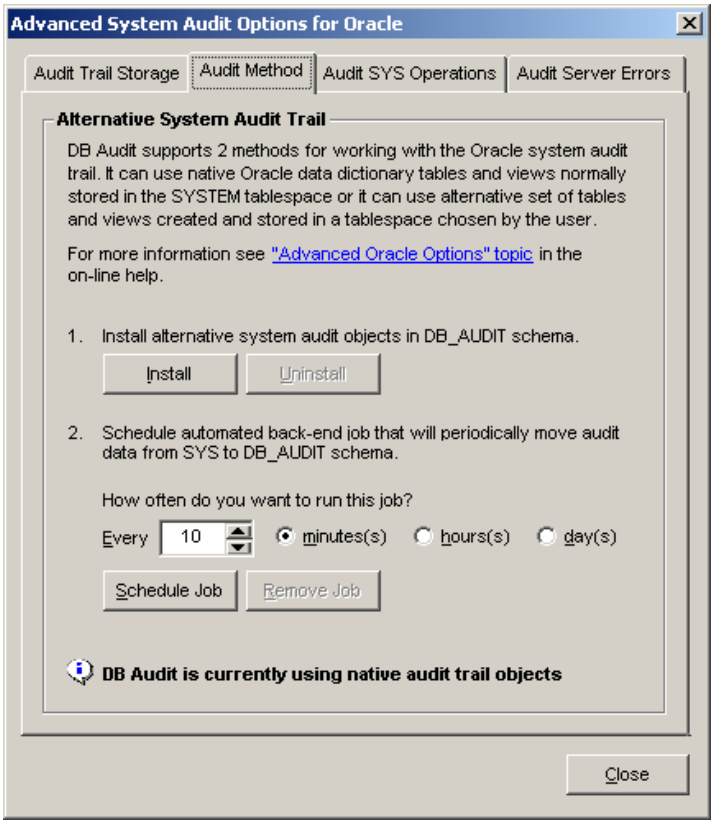
Regardless of the method used to transfer the table, all existing system audit trail data is preserved in all versions of Oracle.

5.

Audit Method and Storage

DB Audit supports two methods for working with the Oracle system audit trail. You can use either of the following:

- The native Oracle data dictionary tables and views, SYS.AUD$ and DBA_AUDIT_..., normally stored in the SYSTEM tablespace
- An alternative set of tables and views created in the DB_AUDIT schema and stored in a user specified tablespace.



To install and switch to the alternative set of audit objects:

1. Start the DB Audit GUI console and connect to the database.

2. Click the **System Audit** menu, then click the **Advanced Options** menu. The Advanced Options dialog displays.

3. Activate the **Audit Method** tab page.

4. Click the **Install** button to install all the required objects in the DB_AUDIT schema. All objects are installed in the default tablespace selected for DB_AUDIT database user. If the DB_AUDIT schema and user do not exist, DB Audit prompts you to select the default tablespace and options which it will use to create the required user. For more information on this process, see Back-end Installation topic in CHAPTER 15 and CHAPTER 4, Data Change Auditing.

5. Choose scheduling options for the system audit data transfer job. This job will periodically move data from the system audit trail table SYS.AUD$ to the alternative DB_AUDIT.AUD$ audit trail table.

6. Click the **Schedule** button to install the data transfer job.

    **Tip**: The practice of using alternative audit trail objects can be used as a solution for the space usage problem in the SYSTEM tablespace; however, this method may lead to high fragmentation of the SYSTEM tablespace if the amount of the generated audit data is very large. For additional information about this issue read the **Moving Oracle native system audit trail table out of the SYSTEM tablespace** section earlier in this chapter.

   During the installation step, your existing system audit trail data is automatically transferred to the alternative tables.

   To uninstall the alternative objects, click the **Uninstall** button and then remove the data transfer job. All existing audit trail data in the alternative tables is lost when the objects are uninstalled.

Auditing Privileged Users connected as SYSDBA or SYSOPER

Although Oracle's audit facilities are very sophisticated, they don't cover all user activities. For example, before Oracle 9iR2, it was not possible to audit the actions of privileged users such as SYS or users connecting "as SYSDBA."

When a user with SYSDBA or SYSOPER privileges connects to the database, actions such as database shutdown, startup, and configuration changes are expected to be performed for administrative reasons only. These actions are assumed not to require auditing and therefore the Oracle auditing mechanism does not pick them up.

Starting with Oracle release 9iR2, the database provides full audit capability for the SYS account. In all versions prior to 9iR2, the only limited auditing option available is the small audit file that Oracle creates by default in the directory $ORACLE_HOME/RDBMS/audit (in a Unix environment), or as a brief audit record written to the Windows NT application Event Log in Windows environments. This file is created every time a user attempts to connect internally through server manager (svrmgrl) or through the now current method of connecting as SYSDBA or SYSOPER. A separate file is written for every connection, making it very difficult, if not impossible, to continuously analyze the audit trail data

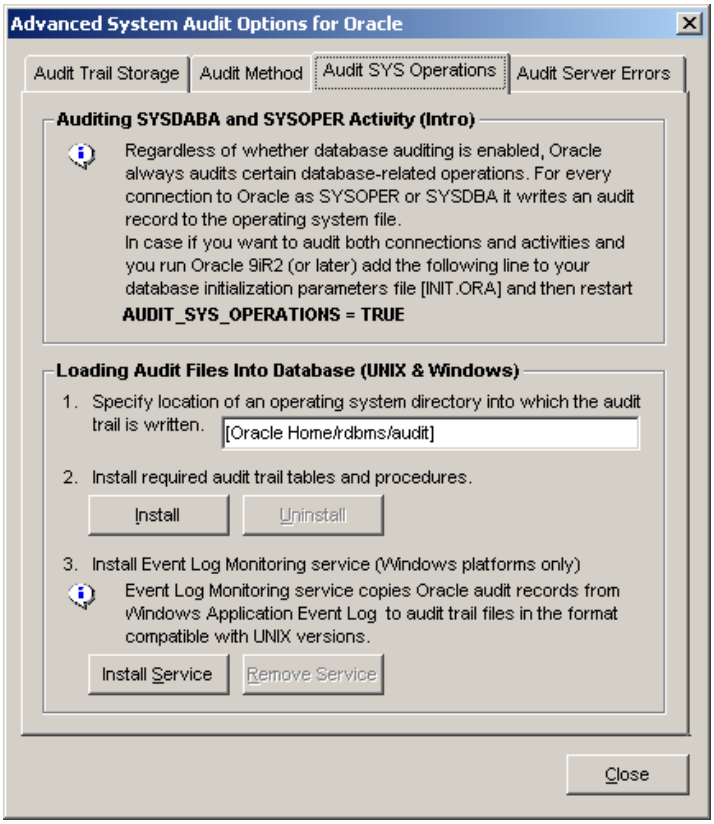
and monitor security breaches.

DB Audit provides methods for monitoring Oracle audit files and loading them into a database table, making such audit data available for reporting and monitoring. These methods fill the gap in Oracle database auditing mechanisms. The audit data loaded into a table can be analyzed using DB Audits' built-in User Activity (Sys Admins) Report or used in custom reports and alerts.

If you run Oracle 9iR2 or later and you want to audit both connections and activities, add the following line to your database initialization parameters file [INIT.ORA], then restart the database:

AUDIT_SYS_OPERATIONS = TRUE

This parameter is independent of the setting in the AUDIT_TRAIL parameter. It is a static parameter that cannot be set using the ALTER SYSTEM command, since the database must be bounced for it to take effect.



To install auditing of SYS-type operations:

1. Start DB Audit GUI console and connect to the database.

2. Click the **System Audit** menu, then click the **Advanced Options** menu. The Advanced Options dialog displays.

3. Activate the **Audit SYS Operations** tab page.

4. Specify the system directory to which the audit trail will be written. You must specify an absolute directory name; you cannot use system environment variables pointing to Oracle home. Also, do not include a trailing slash in the directory name.

Tip: In Unix environments, the directory name is case-sensitive. Make sure to enter it correctly.

Important Notes: The specified audit directory must be listed in the initialization parameter UTL_FILE_DIR. You must add it to your database initialization parameters file [INIT.ORA] and restart the database before you install Audit SYS Operations procedures. Here is an example line from [INIT.ORA] file:

util_file_dir = c:\oracle\mydb\rdbms\audit

If you already have the UTIL_FILE_DIR parameter in your [INIT.ORA] parameters file that references a different directory, you can add additional UTIL_FILE_DIR parameters to the file. To specify multiple directories, include a separate UTL_FILE_DIR line for each directory, but make sure these are on consecutive lines; otherwise only the last directory will be accessible.

5. Click the **Install** button to install all the required objects in the DB_AUDIT schema. All objects are installed in the default tablespace selected for DB_AUDIT database user. If the DB_AUDIT schema and user do not exist, DB Audit prompts to select the default tablespace and options which it will then use to create the required user. For more information on this process see Back-end Installation topic in CHAPTER 15 and also see CHAPTER 4, Data Change Auditing.

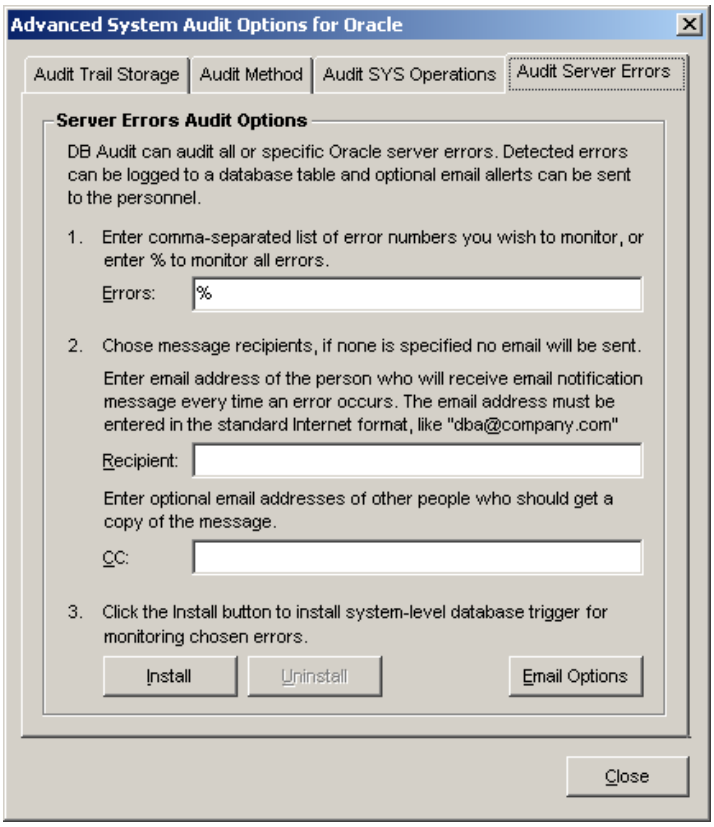
6. If you are running an Oracle database on a Windows server, install the DB Audit Event Log Monitoring service on Windows. The Event Log Monitoring service copies Oracle audit records from the Windows Application Event Log to audit trail files in a format compatible with Unix versions.
   Important: The DB Audit Event Log Monitoring service must be installed on the computer running your Oracle server, not the computer running DB Audit GUI. You must install DB Audit console on the server computer in order to install the Event Log Monitoring service.

The data loaded into a table can be then analyzed using DB Audits' built-in User Activity (Sys Admins) Report.

Tip: The audit trail data is written to the DB_AUDIT.SYSFILE_AUD$ table. DB Audit provides built-in reports for analyzing and reporting this data. For details see User Activity (Sys Admins) Report topic in CHAPTER 15.

Auditing Oracle Database Errors

In Oracle versions 8i and later, DB Audit can take advantage of Oracle system-level triggers to audit Oracle server errors. DB Audit can audit all errors or only specific Oracle server errors. Detected errors can be logged to a database table and can be configured to generate email alerts sent to responsible personnel.



To install procedures for automatic capturing of database errors:

1. Start the DB Audit GUI console and connect to the database.

2. Click the **System Audit** menu, then click the **Advanced Options** menu. The Advanced Options dialog displays.

3. Activate the **Audit Server Error** tab page.

4. If you want to audit specific Oracle errors, enter the Oracle error numbers as a comma-separated list. To capture and audit all errors, enter the per cent sign (%).

5. If you want to implement email alerts, specify email the alert recipients. If no recipients are specified, no email notifications will be sent.

   Important: The email procedure and settings must be installed and configured prior to installing database error auditing. If you haven't configured your email settings yet, click the Email Options button to install the email procedure. For more information, see Setting Email Alerts topic in CHAPTER 4.

6. Click the **Install** button to install all required objects in the DB_AUDIT schema. All objects are installed in the default tablespace chosen for DB_AUDIT database user. If the DB_AUDIT schema and user do not exist yet, DB Audit prompts you to select the default tablespace and options and in which will create the required user. For more information on this process see Back-end Installation topic in CHAPTER 15 and also see CHAPTER 4, Data Change Auditing.

Tip: The audit trail data for database errors is written to the DB_AUDIT.SERVER_ERRORS table. DB Audit provides built-in reports for analyzing and reporting this data. For details see Database Errors Report topic in CHAPTER 15.

Microsoft SQL Server: Configuring System Audit Options

Microsoft SQL Server: Enabling system audit

In order to manage audit options for SQL Server, you must connect to your database as SA user who also has administrative privileges on the computer running the SQL Server service.

Important Note: In order to enable system audit in SQL Server, DB Audit installs several stored procedures in the DBO schema in the master database as well as several tables and procedures in a user-selected repository database within the DB_AUDIT schema. DB Audit Management Console automatically installs all these objects and creates the necessary user and schema when you click the Enable System Audit button on the System Audit Configuration screen.

If the DB Audit Management Console is running locally on the SQL Server computer, it automatically deploys the **xp_dbaudit.dll** file that is required for the auditing services.

If the DB Audit Management Console is running remotely, you must copy the **xp_dbaudit.dll** file manually from the DB Audit Management Console computer to the SQL Server computer. Look for this file in the MSSQL subfolder in the DB Audit Management Console installation folder. Copy this file to the BINN subfolder of your SQL Server instance. Make sure to pick the correct version of **xp_dbaudit.dll** file. There are three different versions of this file:

- version for x86 32-bit systems
- version for x86 64-bit systems
- version for Intel Itanium based IA 64-bit systems

Enabling System Audit and Selecting the Audit Repository Location

When you first use DB Audit to configure system audit settings and you click the **Enable System Audit** button, DB Audit automatically installs all required database objects including several regular and extended stored procedures. It also installs the following audit trail tables and configuration tables that are used internally to save audit settings and audit results:

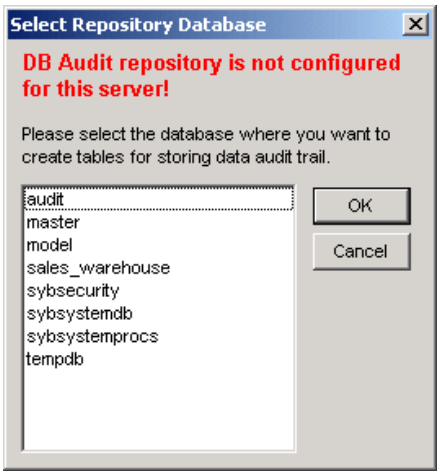
DB_AUDIT.SYS_AUDIT_TRAIL	Used to store audit results (also called system audit trail)
DB_AUDIT.SYS_AUDIT_CONFIG	Used to store audit settings. Do not modify data in this table directly as this may lead to system audit malfunction.

Both tables are created in the same DB_AUDIT schema.

To create the DB_AUDIT schema, SQL Server requires that a user with the same name exists in the database. When you use DB Audit to enable the auditing, it automatically creates the required user and, if required, a matching login name.

Important Note:

Do not use the created DB_AUDIT user login to connect to your database. This user is created for the sole purpose of maintaining a separate audit schema where all audit objects and data are stored. You should use the SA account to configure the system audit settings using DB Audit GUI.



The first time you enable the system audit, DB Audit prompts you to choose a database where DB Audit will create the DB_AUDIT schema objects and store audit trail data.

Although not required, it is highly recommended that you choose a separate database dedicated to storing the audit trail data. By using a separate database, you can greatly reduce overall system maintenance and reduce system and storage overhead requirements for audit data. In addition, the production database is also separate from the audit data and thus performance and space management is simplified. Audit data can also be secured easily. If you wish to use a separate database but don't have it yet, cancel the **Select Repository Database** dialog, create a new database using SQL Server Enterprise Manager and then return to the DB Audit and repeat the system audit configuration process.

Important Notes:

1. Make sure the database selected for the repository has "Select into/bulk copy" option enabled. If it does not, either use available database graphical management tools (such as SQL Server Enterprise Manager) or execute **sp_dboptions** system procedure manually to enable the "Select into/bulk copy" option. To enable this option using **sp_dboptions** system procedure run the following SQL command, replacing *'repository db name'* parameter with the real repository database name.

   ```
   exec sp_dboptions 'repository db name', 'select into/bulkcopy', 'on'
   go
   ```

2. If the repository database differs from the database containing the audited table make sure database users who change data in the audited tables also exist in the repository database. To add a user to the repository database you can use available database graphical management tools (such as SQL Server Enterprise Manager) or execute **sp_adduser** system procedure. To add a user using **sp_adduser** system procedure run the following SQL commands, replacing *'repository db name'* parameter with the real repository database name and replacing *'login'* and *'user'* parameters with the real login name and database user name.

   ```
   use 'repository db name'
   go
   exec sp_adduser 'login', 'user'
   go
   ```

Alternative Method to Start System Audit

Normally the system auditing process starts automatically with SQL Server when SQL Server service

starts. In case you want to start the auditing manually, perhaps immediately after the initial installation you can use the following method:

SQL Server 2000:

1. Start SQL Query Analyzer and connect to the database server.
2. Type and execute the following command

   ```
   exec master.dbo.sp_dbaudit_startup
   ```

3. Using Windows Task Manager kill the SQL Query Analyzer process (do not try to cancel the query or exit SQL Query Analyzer normally, because the process will not).

SQL Server 2005/2008:

1. Start SQL Server Management Studio and connect to the database server.
2. Open new query window.
3. Type and execute the following command

   ```
   exec master.dbo.sp_dbaudit_startup
   ```

4. Using Windows Task Manager kill the SQL Server Management Studio process (do not try to cancel the query or exit SQL Server Management Studio normally, because the process will not).

Microsoft SQL Server: Disabling system audit

1. Click **System Audit/Set Audit Options** command from the DB Audit Expert menu to launch system audit configuration screen.
2. Click the **Disable System Audit** button displayed on the bottom of the system audit configuration screen.

Microsoft SQL Server: Setting system audit options

Click **System Audit/Set Audit Options** command from the DB Audit Expert menu to launch system audit configuration screen.

The system audit configuration screen consists of 3 tab pages:

1. **Operations** – use options available on this page to select when and which operations and events to audit.
2. **Filters** – use options available on this page to configure audit operation filters.

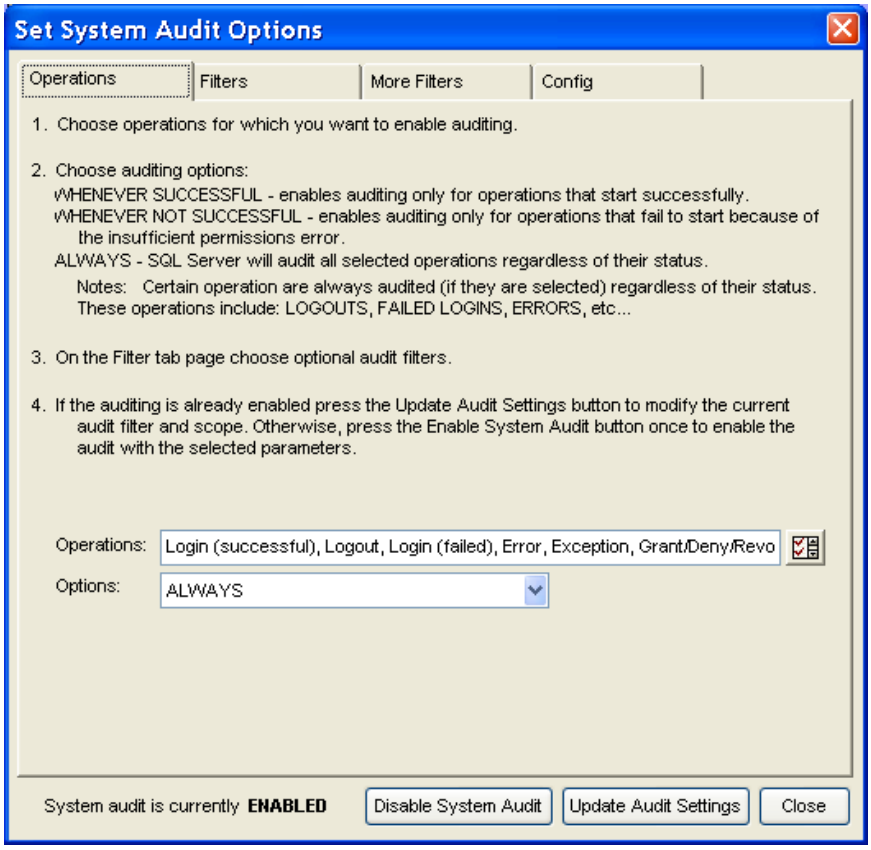
Tip: Filters can be applied to the selected audit events to limit the results of the auditing to those events generated by a specific user or application. Filters can also be set to look for specific statements or specific database objects.

3. **Config** – use options available on this page to configure audit trail behavior.

Tip: Options on this page affect how DB Audit monitors SQL Server events and records audit trail data.

Selecting Operations

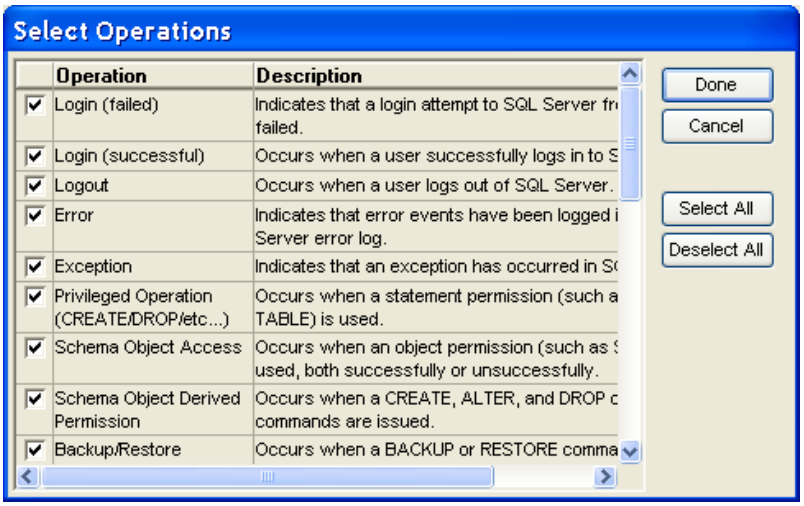
The following screenshot demonstrates typical selection for audit Operations settings.



To select which Operations to audit:

1. Activate the **Operations** tab page. Choose operations for which you want to enable auditing.

   Click the **Lookup** button  displayed on the **Operations** line to open the **Select Operations** dialog.



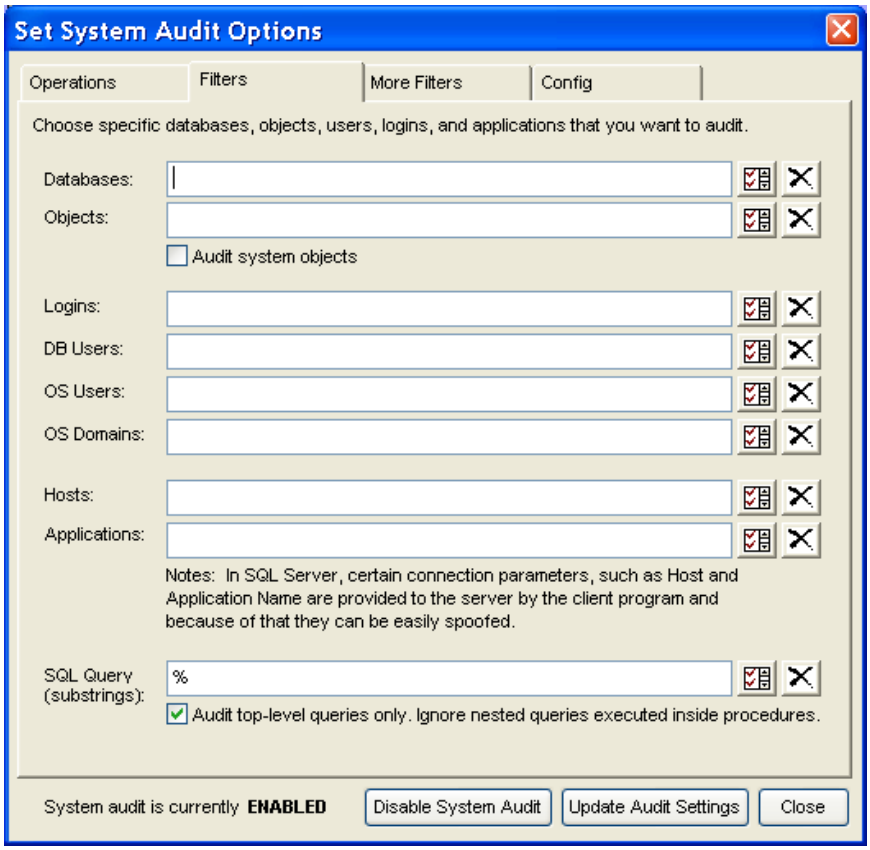
You can select one or more operations by placing a checkmark into the left-most column. To quickly select all listed operations click the **Select All** button. To quickly deselect all previously selected operations click the **Deselect All** button. Click the **OK** button to return to the previous screen with all selected operations. To cancel the dialog without making any changes click the **Cancel** button.

2. Choose WHENEVER SUCCESSFUL option to enable auditing only for **authorized database operations**. Choose WHENEVER NOT SUCCESSFUL option to enable auditing only for failed operations, in other words, auditing of attempts to issue **non-authorized commands**. If you select ALWAYS or leave the **Options** blank, SQL Server will audit all database operations regardless of success or failure.

3. If you don't want to setup any audit event filters, click the **Enable System Audit** button or **Update Audit Settings** button (whichever is enabled) to save the new settings and if necessary startup the auditing process. Use the **Update Audit Settings** button if the system audit is already installed and enabled. Use the **Enable System Audit** button if the system audit is not yet installed and you are using the audit configuration settings screen first time. This button will do both things: save chosen audit settings and install DB Audit procedures and catalog tables.

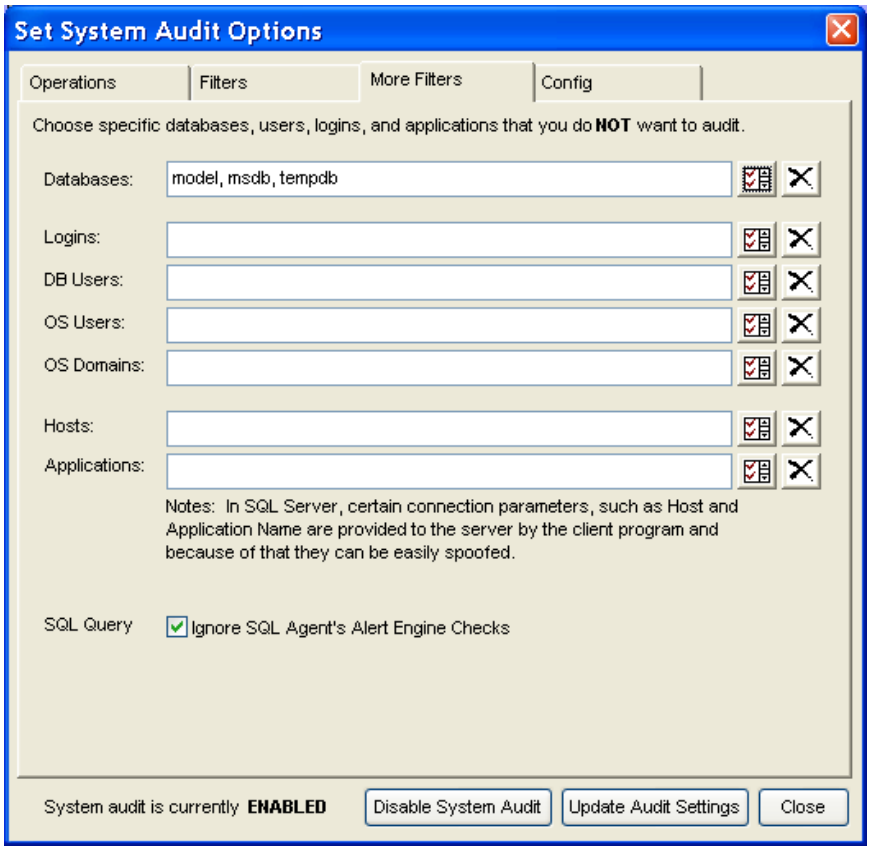
To select audit filters:

1. Activate the **Filters** tab page. Choose appropriate **Inclusive** audit filters. The following screenshot demonstrates typical selection for audit filters settings.



Multiple filtering parameters can be selected. All together, they create one complex filter applied to the audit process. Logical AND operations are used to join different parameters together. For example, if you select *master* and *msdb* databases for the **Databases** parameter and also select *MS SQLEM* and *SQL Query Analyzer* for the **Applications** parameter, then the audit trail will contain only records related to user activities using SQL Server Enterprise Manager and SQL Query Analyzer programs in *master* and *msdb* databases.

You can also use the **More Filters** Page to choose **Exclusive** filters. It is typical for example to exclude events occurring in *tempdb* from the auditing.



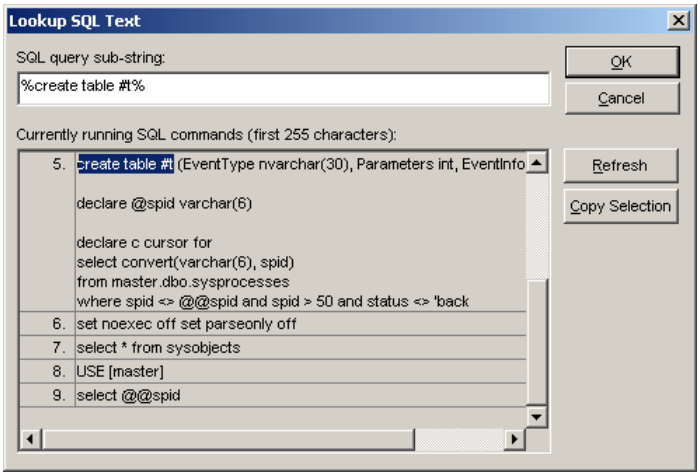
The following filtering parameters are supported:

Databases	Use this parameter to limit the auditing to specific databases. Multiple databases can be selected for the filter. Click the **Lookup** button displayed on the **Databases** line to open the **Lookup Database Names** dialog. The dialog lists all existing databases. Select one or more databases by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected databases into the Databases parameter. Click the **Delete** button displayed on the **Databases** line to clear this parameter.
Objects	Use this parameter to limit the auditing to specific schema objects. Multiple objects from the same schema can be selected for the filter. Click the **Lookup** button displayed on the **Objects** line to open the **Lookup Object Names** dialog. When the dialog appears on the screen select the required schema using the **Schema** drop-down list. This will populate the object names list with all existing objects in the selected schema. Select one or more objects by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter

	selected objects into the Objects parameter. Click the **Delete** button  displayed on the **Objects** line to clear this parameter.
Audit System Objects	Check this box to enable auditing of access to system objects. By default if this option is not checked DB Audit will only audit access to non-system objects
Logins	Use this parameter to limit the auditing to specific logins. Multiple logins can be selected for the filter. Click the **Lookup** button  displayed on the **Logins** line to open the **Lookup Login Names** dialog. The dialog lists all existing database logins. Select one or more logins by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected logins into the Logins parameter. Click the **Delete** button  displayed on the **Logins** line to clear this parameter.
DB Users	Use this parameter to limit the auditing to specific database users. Multiple users can be selected for the filter. Click the **Lookup** button  displayed on the **DB Users** line to open the **Lookup Database User Names** dialog. The dialog lists all existing database users from all databases. Select one or more users by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected names into the DB Users parameter. Click the **Delete** button  displayed on the **DB Users** line to clear this parameter.
OS Users	Use this parameter to limit the auditing to specific Operation System users (in other words, network user names). Multiple users can be selected for the filter. Click the **Lookup** button  displayed on the **OS Users** line to open the **Lookup Network User Names** dialog. **Important Notes:** The dialog lists names of all network users currently connected to the database. If you do not see the required name listed there ask that user to run any database application and then close and reopen the lookup dialog. Select one or more users by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected names into the OS Users parameter.

	Click the **Delete** button  displayed on the **OS Users** line to clear this parameter.
OS Domains	Use this parameter to limit the auditing to specific network domains (in other words, limit auditing to network users authenticated using chosen domains only). Multiple domains can be selected for the filter. Click the **Lookup** button  displayed on the **OS Domains** line to open the **Lookup Network Domains** dialog.  **Important Notes:** The dialog lists names of all network domains whose users are currently connected to the database. If you do not see the required name listed there ask user form that domain to run any database application and then close and reopen the lookup dialog. Select one or more domains by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected domain names into the OS Domains parameter. Click the **Delete** button  displayed on the **OS Domains** line to clear this parameter.
OS Domains	Use this parameter to limit the auditing to specific network domains (in other words, limit auditing to network users authenticated using chosen domains only). Multiple domains can be selected for the filter. Click the **Lookup** button  displayed on the **OS Domains** line to open the **Lookup Network Domains** dialog.  **Important Notes:** The dialog lists names of all network domains whose users are currently connected to the database. If you do not see the required name listed there ask user form that domain to run any database application and then close and reopen the lookup dialog. Select one or more domains by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected domain names into the OS Domains parameter. Click the **Delete** button  displayed on the **OS Domains** line to clear this parameter.
Hosts	Use this parameter to limit the auditing to specific computers from which users connect to the database.  **Important Notes:** in SQL Server certain connection parameters such as Host and Application Name are application supplied and because of that they can be easily spoofed. Some applications even allow users to specify Host and Application Names before establishing a

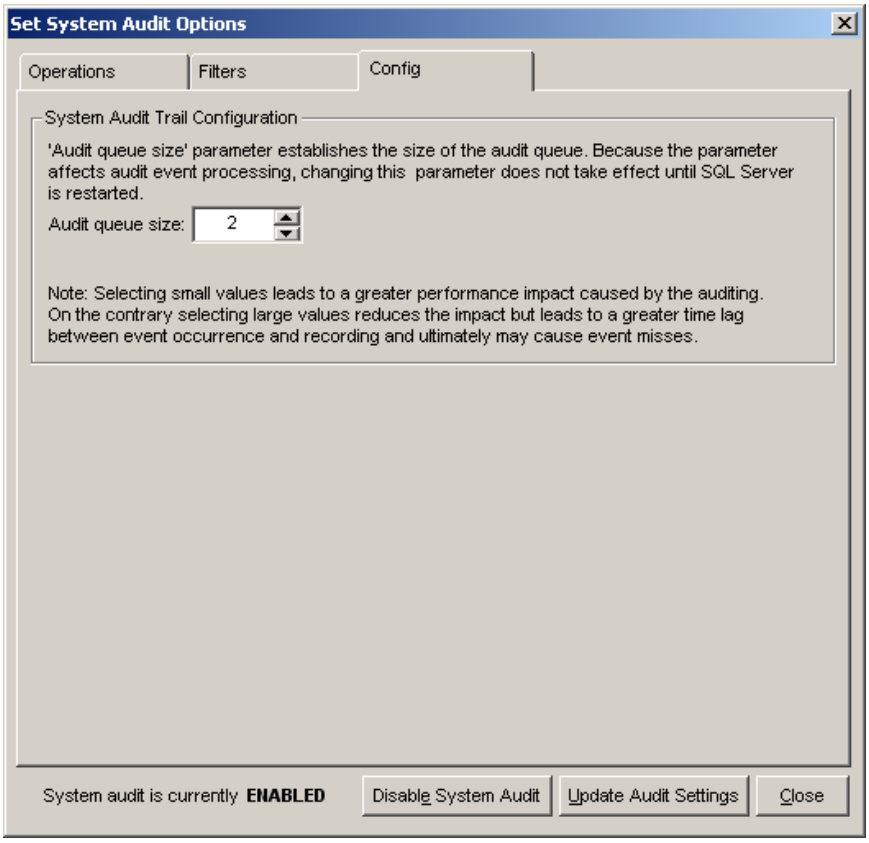
	new database connection or using global application settings. Multiple hosts can be selected for the filter. Click the **Lookup** button  displayed on the **Hosts** line to open the **Lookup Hosts** dialog. **Important Notes:** The dialog lists names of all **computers** whose users are currently connected to the database. If you do not see the required name listed there run any database application on the required computer and then close and reopen the lookup dialog. Select one or more hosts by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected host names into the Hosts parameter. Click the **Delete** button  displayed on the **Hosts** line to clear this parameter.
Application Names	Use this parameter to limit the auditing to specific database applications. **Important Notes:** in SQL Server certain connection parameters such as Host and Application Name are application supplied and because of that they can be easily spoofed. Some applications even allow users to specify Host and Application Names before establishing a new database connection or using global application settings. Multiple applications can be selected for the filter. Click the **Lookup** button  displayed on the **Application Names** line to open the **Lookup Applications** dialog. **Important Notes:** The dialog lists names of all database applications currently connected to the database. If you do not see the required name listed there run that database application and then close and reopen the lookup dialog. Select one or more hosts by placing a checkmark into the left-most column. Click the OK button to close the lookup dialog and enter selected application names into the Application Names parameter. Click the **Delete** button  displayed on the **Application Names** line to clear this parameter.
SQL Query	Use this parameter to limit the auditing to database queries whose text contains specific substrings. Only one substring can be selected for the filter. Standard LIKE comparison type is used to filter SQL queries. Substrings can include standard % wildcards used in T-SQL. Click the **Lookup** button  displayed on the **SQL Query** line to open the **Lookup SQL Queries** dialog.

	Important Notes: For your convenience the dialog lists all currently running SQL queries.  You can use the following methods to enter the required substring: In the **SQL query substring** field type the substring. You can enter any text you want even if there are no matching SQL query listed below the field. *or* In the **Currently running SQL commands** list highlight the required substring and then click the **Copy Selection** button to copy the highlighted text to the **SQL query substring** field. Click the OK button to close the lookup dialog and enter selected substring into the SQL Query parameter. Click the **Delete** button  displayed on the **SQL Query** line to clear this parameter.
Audit Top-level Queries Only	Use this parameter to limit the auditing to top-level database queries only. If this option is checked all other queries executed from stored procedures and triggers are ignored and not recorded in the audit trail. For example, a user runs application APP that executes stored procedures SP. The SP procedure runs 10 different SQL commands including an UPDATE for some user table with a trigger TR which in turn runs another set of commands. If this option is not checked DB Audit will record *EXEC SP* command and all other SQL commands executed within the stored procedure and trigger. If this option is checked then only *EXEC SP* command is recorded.

2. If you have already selected operations for which you want to enable the auditing, click the **Enable System Audit** button or **Update Audit Settings** button (whichever is enabled) to save the new settings and if necessary startup the auditing process. Use the **Update Audit Settings** button if the system audit is already installed and enabled. Use the **Enable System Audit** button if the system audit is not yet installed and you are using the audit configuration settings screen first time. This button will do both things: save chosen audit settings and install DB Audit procedures and catalog tables.

To configure advanced audit options:

1. Activate the **Config** tab page. Specify the **Audit Queue Size** parameter. The following screenshot demonstrates typical selection for advanced settings.



Audit queue size parameter establishes the size of the audit queue. Because this parameter affects audit event processing, changing this parameter does not take effect until SQL Server is restarted.

Important Notes: Small values lead to a greater database performance impact caused by the auditing. On the contrary large values reduce the impact but lead to a greater time lag between event occurrence and recording and ultimately may cause event misses.

2. If you have already selected filters and operations for which you want to enable the auditing, click the **Enable System Audit** button or **Update Audit Settings** button (whichever is enabled) to save the new settings and if necessary startup the auditing process. Use the **Update Audit Settings** button if the system audit is already installed and enabled. Use the **Enable System Audit** button if the system audit is not yet installed and you are using the audit configuration settings screen first time. This button will do both things: save chosen audit settings and install DB Audit procedures and catalog tables.

Sybase SQL Server and ASE: Configuring System Audit Options

Sybase: Enabling system audit

To manage audit options for Sybase databases you must connect to your database as SA user.

> **Important Note: Sybase auditing requires that the audit system is installed on the server.** The audit system consists of the *sybsecurity* database and several system procedures and configuration parameters. If you do not have these components installed, refer to your Sybase System Administration Guide for instruction on how to run the **auditinit** installation procedure.

To enable the system audit:

1. Click **System Audit/Set Audit Options** command from the DB Audit Expert menu to launch the System Audit Configuration screen.

2. Click the **Enable System Audit** button displayed at the bottom of the System Audit Configuration screen.

Sybase: Disabling system audit

To disable system audit:

1. Click the **System Audit/Set Audit Options** command on the DB Audit Expert menu to launch the System Audit Configuration screen.

2. Click the **Disable System Audit** button displayed at the bottom of the System Audit Configuration screen.

Sybase: Setting system audit options

Click the **System Audit/Set Audit Options** command on the DB Audit Expert menu to launch System Audit Configuration screen.

DB Audit GUI presents ASE auditing options organized in four groups:

1. **Global** – These options can be used to track the occurrence of server-specific SQL statements and privilege usage in user sessions. You can track the occurrence of a specific SQL statement or of all SQL statements authorized by a particular system privilege.

   Server-specific auditing operations apply immediately to current and subsequent sessions. Both successful and unsuccessful operations can be tracked.

2. **Database** – These options can be used to track the occurrence of database-specific SQL statements and privilege usage in user sessions. You can track the occurrence of a specific SQL statement or of all SQL statements authorized by a particular system privilege.

Database-specific auditing operations apply immediately to current and subsequent sessions. Both successful and unsuccessful operations can be tracked.

3. **Schema Objects** – These options can be used to track operations on a specific schema object.

   Auditing operations on schema objects apply immediately to current and subsequent sessions. Both successful and unsuccessful operations can be tracked.

4. **Sessions** – These options can be used to track the occurrence of database logons and logoffs, as well as to track session-level access to database tables and views. Both successful and unsuccessful logons and operations can be tracked.

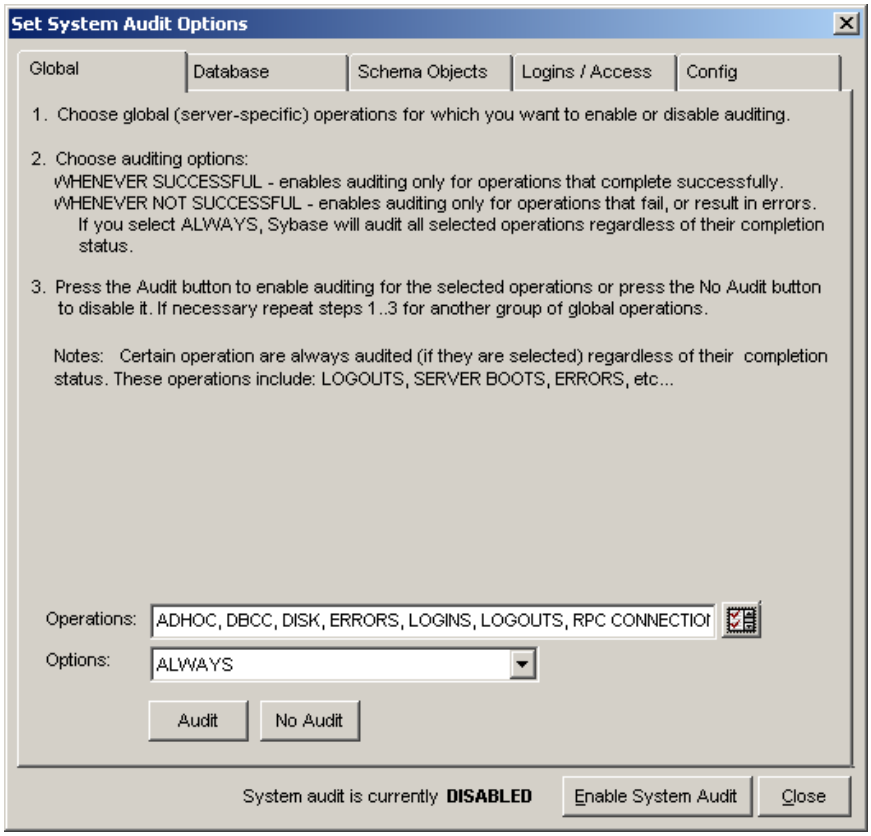
Each group of auditing options is displayed on a separate tab page. DB Audit provides context help and usage instructions directly on the auditing options screen.

Tips:

- You can use **Enabled System Audits** reports from the **Reports** menu to see which system audit options are currently enabled in the database.
- Java stored procedures are considered the same as T-SQL stored procedures for purposes of auditing options.

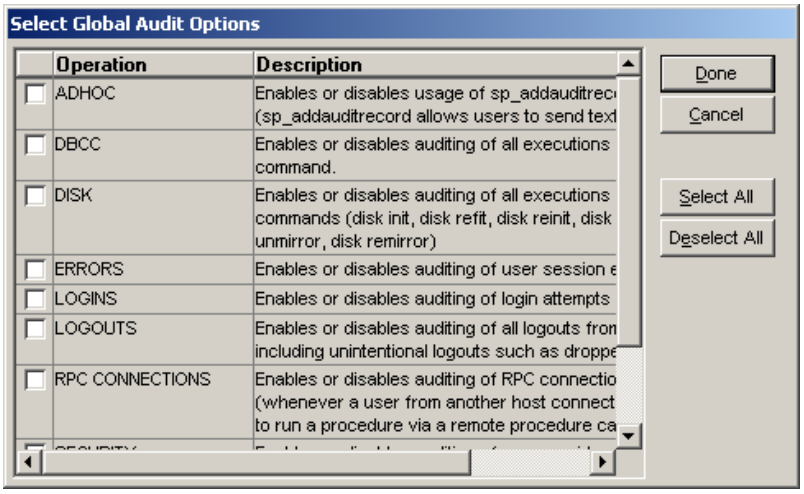
Setting Global Audit Options

The following screenshot demonstrates typical selections for auditing server-specific operations.



To setup auditing for global server-specific operations:

1. Activate the **Global** tab page. Choose specific operations for auditing. Click the **Lookup** button  displayed on the **Operations** line to open the **Select Global Audit Options** dialog.



You can select one or more operations by placing a checkmark into the left-most column. To quickly select all listed options, click the **Select All** button. To quickly deselect all previously selected options, click the **Deselect All** button. Click the **OK** button to return to the previous screen with all selected options. To cancel the dialog without making any changes, click the **Cancel** button.

If no specific operations are selected, DB Audit will default to the AUDIT ALL option, meaning that all supported operations will be audited.

2. Using the **Options** drop-down list at the bottom of the screen, select either the AUDIT ALL or AUDIT ONLY SUCCESSFULE OR UNSUCCESSFUL OPERATIONS option..

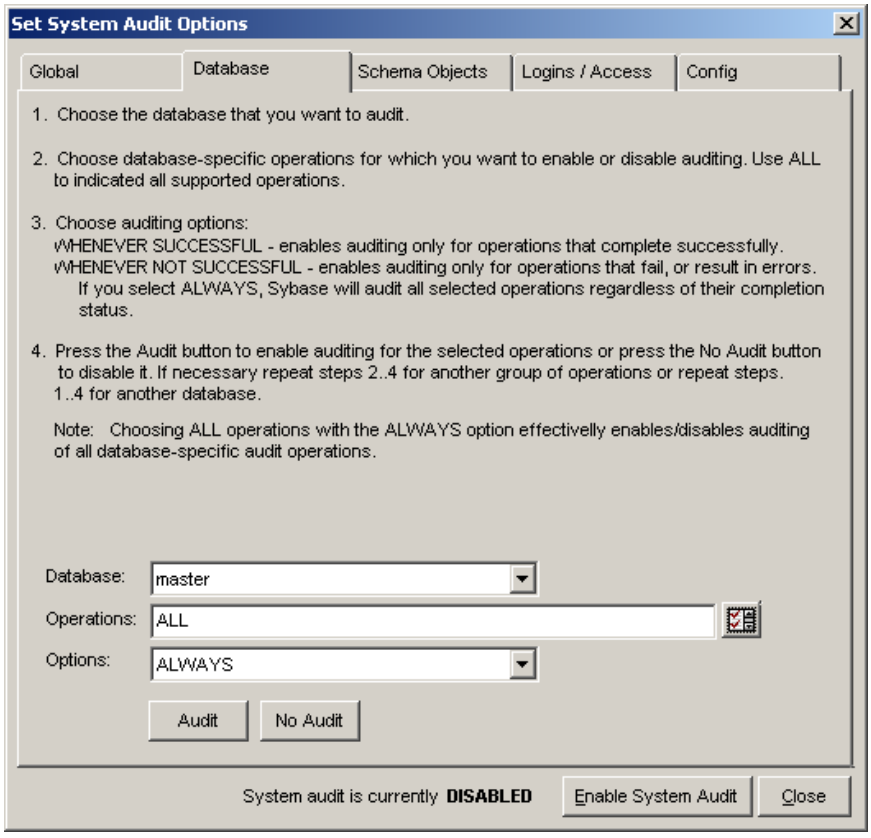
   Choose the WHENEVER SUCCESSFUL option to enable auditing only for operations that complete successfully. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for operations that fail or that generate errors. If you select ALWAYS or leave the **Options** field blank, ASE will audit operations regardless of success or failure.

3. Press the **Audit** button to enable auditing of the selected operations and options, or press the **No Audit** button to disable it. If necessary repeat steps 1 through 3 for another group of operations.

   Tip: You can select different audit options for different group of operations. For example, you can audit all DBCC executions, including successfully completed and failed ones, and at the same time audit only failed logons.

Setting Database Audit Options

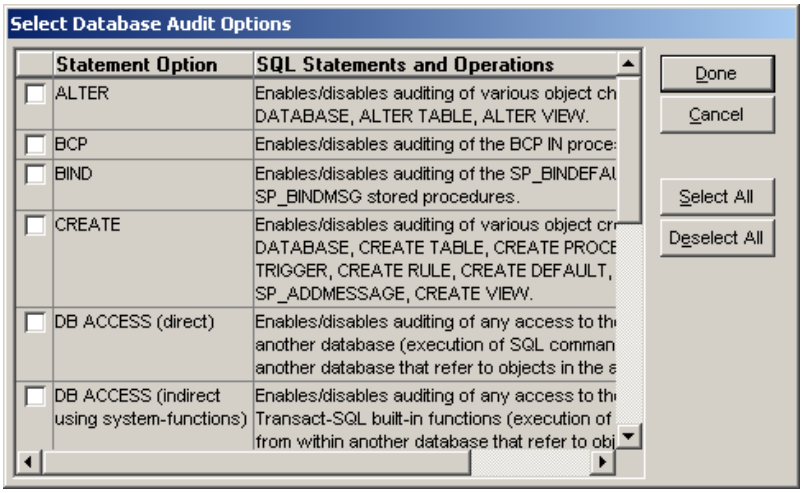
The following screenshot demonstrates typical selections for database-specific operations auditing.



To setup auditing for database-specific operations:

1. Activate the **Database** tab page. Choose the database you want to audit.

2. Choose specific operations for auditing. Click the **Lookup** button  displayed on the **Operations** line to open the Select Database Audit Options dialog.



Select one or more operations by clicking the appropriate checkboxes in the first column. To quickly select all listed options, click the **Select All** button. To quickly deselect all previously selected options, click the **Deselect All** button. Click the **OK** button to return to the previous screen with all selected options. To cancel the dialog without making any changes, click the **Cancel** button.

If no specific operations are selected, DB Audit defaults to the AUDIT ALL option, meaning that all supported operations are audited.

3. Choose whether to audit all or only successful or unsuccessful operations using the **Options** drop-down list at the bottom of the screen.

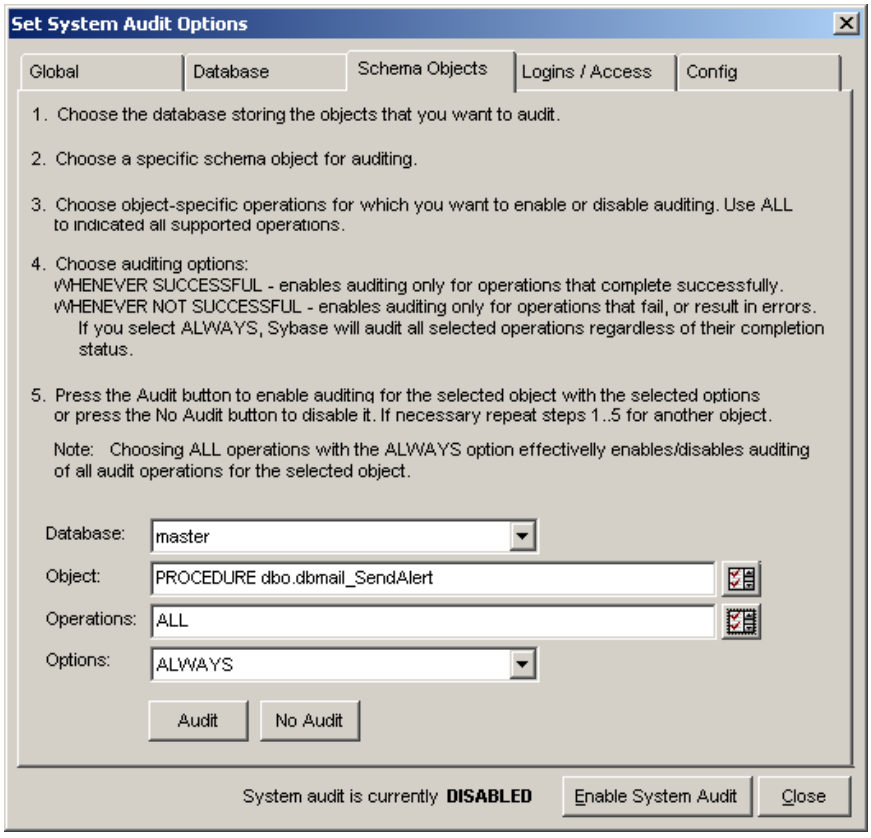
   Choose the WHENEVER SUCCESSFUL option to enable auditing only for operations that complete successfully. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for operations that fail or that generate errors. If you select ALWAYS or leave the **Options** field blank, ASE audit operations regardless of success or failure.

4. Press the **Audit** button to enable auditing of the selected operations, or press the **No Audit** button to disable it. If necessary, repeat steps 1 through 4 for another database or group of operations.

   Tip: You can select different audit options for different databases and groups of operations within the same database. For example, you can audit all DBCC executions in the MASTER database, including successfully completed and failed ones, and at the same time audit only failed BCP commands in the TRADING database.

Setting Schema Object Audit Options

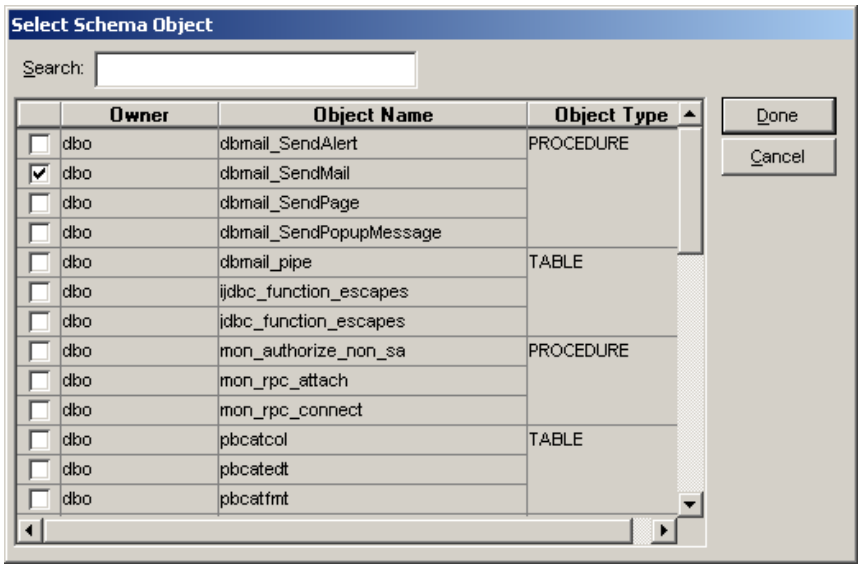
The following screenshot demonstrates typical selections for object-specific operations auditing.



To setup auditing for object-specific operations:

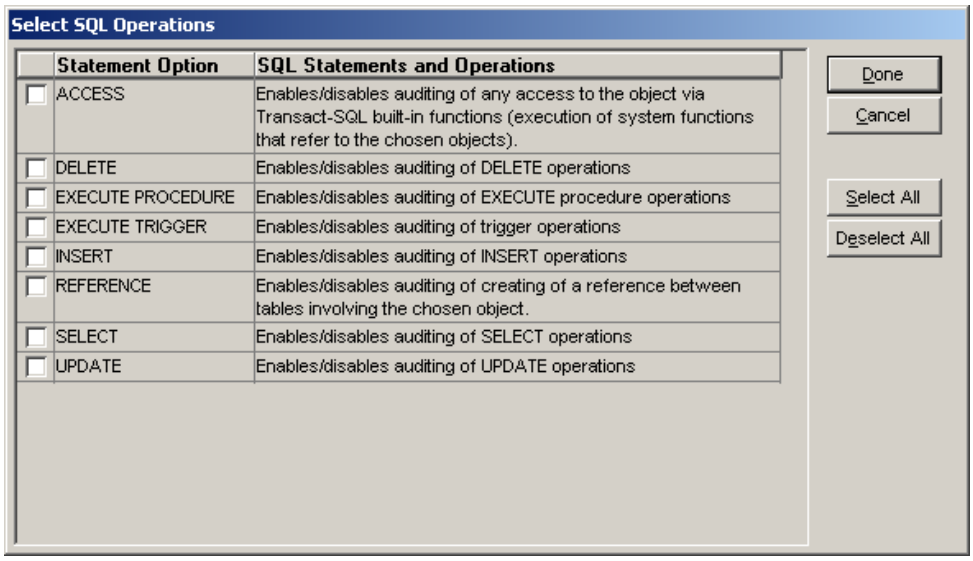
1. Activate the **Schema Object** tab page. Choose the database containing the objects you want to audit.

2. Choose the schema object for which you want to enable or disable certain audit options. Click the **Lookup** button  on the **Operations** line to open the Select Schema Object dialog.



Select the required object by clicking the appropriate checkbox in the first column. Click the **OK** button to return to the previous screen with the selected object. To cancel the dialog without making any changes, click the **Cancel** button.

3. Choose specific operations for auditing. Click the **Lookup** button  on the **Operations** line to open the Select SQL Operations dialog.



Select one or more operations to be audited by clicking the appropriate checkboxes in the first column. To quickly select all listed options, click the **Select All** button. To quickly deselect all previously selected options, click the **Deselect All** button. Click the **OK** button to return to the previous screen with all selected options. To cancel the dialog without making any changes, click the **Cancel** button.

Tip: Different operations can be audited for different object types; however, when selecting options for a specific object, you must be careful to select only operation(s) applicable to that object.

If no specific operations are selected, DB Audit defaults to the AUDIT ALL option, meaning

that all supported operations will be audited.

4. Choose whether to audit all or only successful or unsuccessful operations using the **Options** drop-down lists at the bottom of the screen.

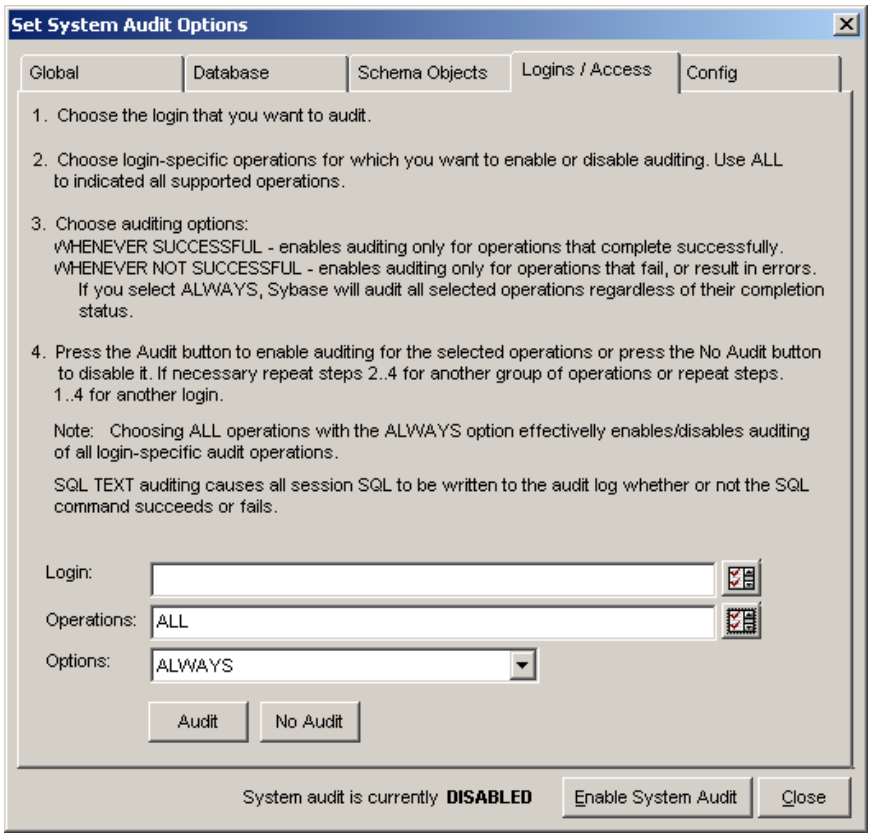
   Choose the WHENEVER SUCCESSFUL option to enable auditing only for operations that complete successfully. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for operations that fail or that generate errors. If you select ALWAYS or leave the **Options** field blank, ASE audits all operations regardless of success or failure.

5. Press the **Audit** button to enable auditing of the selected operations or press the **No Audit** button to disable it. If necessary repeat steps 1 through 5 for another object.

   Tip: You can select different audit options for different database objects. For example, you can audit all EXECUTE PROCEDURE operations for stored procedure MY_PROC, including both successful and failed operations, and at the same time audit only failed DELETE commands for the MY_TABLE table.

Setting Session and Access Audit Options

The following screenshot presents options for session-specific operations auditing.

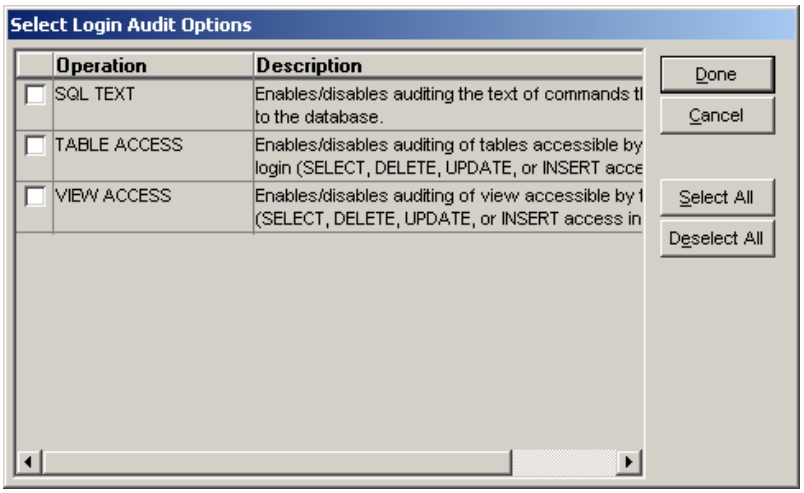


To setup auditing for user sessions:

1. Activate the **Sessions, Access** tab page. Choose the database login you want to audit. Click the **Lookup** button displayed on the **Login** line to open the Lookup Login Names dialog.

The dialog lists all existing database logins. Select the required login by clicking the appropriate checkbox in the first column. Click the **OK** button to return to the previous screen with the selected login name. To cancel the dialog without making any changes click the **Cancel** button.

2. Choose whether to audit logins only, session access to database tables and views, or all logins and session access attempts. Click the **Lookup** button  displayed on the **Operations** line to open the Select Logon Audit Options dialog.



Select the required options by clicking the appropriate checkboxes in the first column. To quickly select all listed options, click the **Select All** button. To quickly deselect all previously selected options, click the **Deselect All** button. Click the **OK** button to return to the previous screen with the selected options. To cancel the dialog without making any changes, click the **Cancel** button.

3. Use the **Options** drop-down lists at the bottom of the screen select either to audit all operations or only successful or unsuccessful operations.

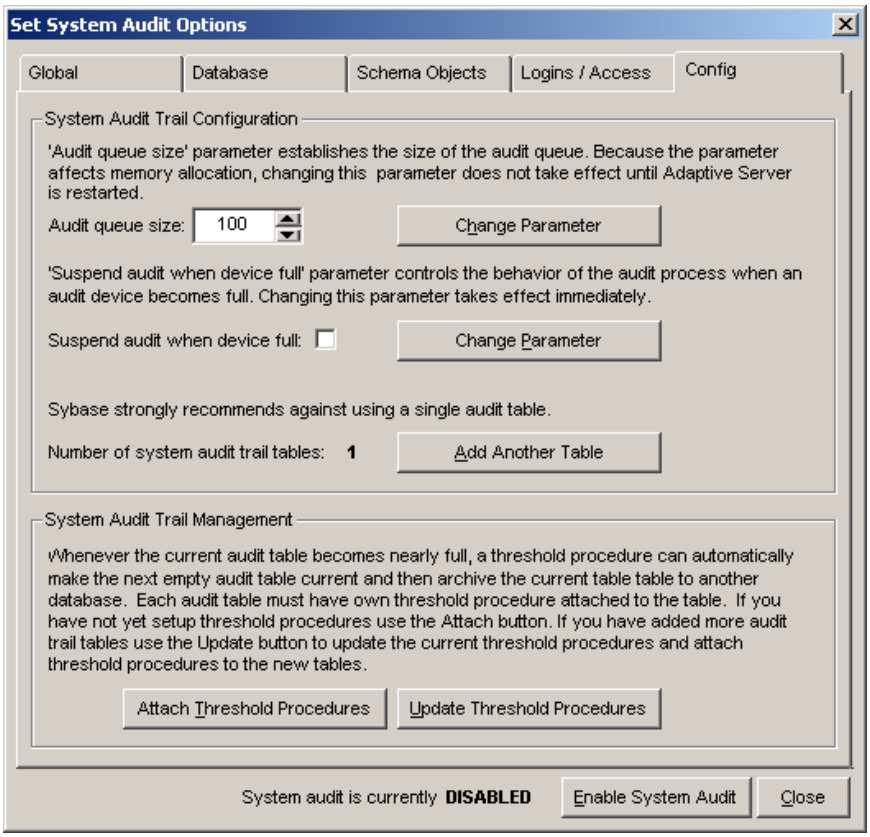
   Choose the WHENEVER SUCCESSFUL option to enable auditing only for operations that complete successfully. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for operations that fail or that generate errors. If you select ALWAYS or leave the **Options** field blank, ASE will audit operations regardless of success or failure.

4. Press the **Audit** button to enable auditing of the selected operations, or press the **No Audit** button to disable it. If necessary repeat steps 1 through 4 for another login.

   Tip: You can select different audit options for different logins.

Setting Audit System Configuration Options and Management Procedures

DB Audit provides a direct interface to the Sybase system audit tuning parameters and configuration. To access this interface, launch system the Audit Configuration screen by clicking the **System Audit/Set Audit Options** command on the DB Audit Expert menu. On this screen, activate the **Config** tab page.



Follow the instructions on the tab page to configure audit system parameters and management procedures. Refer to the *Sybase System Administration Guide* for detailed descriptions of these parameters and procedures.

DB2: Configuring System Audit Options

DB2: Enabling system audit

System-level audit facility is currently supported with DB2 UDB version 6.0 or later running on Unix, Linux or Windows host operation systems.

To manage audit options for DB2, you must connect to the database as a user with SYSADMIN privileges.

> **Important Note:** To enable system audit in DB2, DB Audit installs several tables and procedures in a user-selected repository database within the DB_AUDIT schema. DB Audit GUI can automatically install all of these components when you click the **Enable System Audit** button on the System Audit Configuration screen.

Enabling System Audit and Selecting the Audit Repository Location

The first time you use DB Audit to configure system audit settings and you click the **Enable System**

Audit button, DB Audit automatically installs all required database objects, including several stored procedures and functions. It also installs the following audit trail tables and configuration tables that are used internally to save your audit settings and audit results:

DB_AUDIT.SYS_AUDIT_TRAIL (also called the system audit table)	Used to store audit results
DB_AUDIT.SYS_AUDIT DB_AUDIT.SYS_CHECKING DB_AUDIT.SYS_VALIDATE DB_AUDIT.SYS_OBJMAINT DB_AUDIT.SYS_SECMAINT DB_AUDIT.SYS_SYSADMIN DB_AUDIT.SYS_CONTEXT	Used internally by DB Audit procedures for loading audit trail data into the DB_AUDIT.SYS_AUDIT_TRAIL table. **Do not modify data in these table directly as this may lead to system audit malfunction.**

DB_AUDIT.SYS_AUDIT_TRAIL – This table is used to store audit results (also called system audit trail)

DB_AUDIT.SYS_AUDIT, DB_AUDIT.SYS_CHECKING, DB_AUDIT.SYS_VALIDATE, DB_AUDIT.SYS_OBJMAINT, DB_AUDIT.SYS_SECMAINT, DB_AUDIT.SYS_SYSADMIN, and DB_AUDIT.SYS_CONTEXT – All these worktables are used internally by DB Audit procedures for loading audit trail data into the DB_AUDIT.SYS_AUDIT_TRAIL table. Do not modify data in these table directly as this may lead to system audit malfunction.

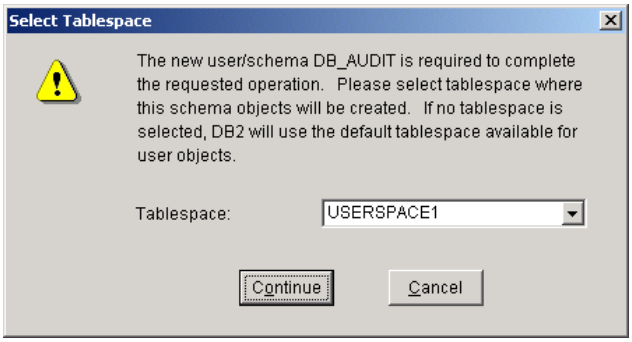
All tables are created in the same DB_AUDIT schema.

In order to create the DB_AUDIT schema, DB2 requires that a user with the same name exists in the database. When you use DB Audit to enable auditing, it automatically creates the required user and, if necessary, a matching login name. DB Audit attempts to use the default password for the new DB_AUDIT user / login. If you want to change this password before the installation, or if the password does not pass your database system password complexity rules, use the Options screen to change the default password value. See Options topic for more details.

> **Important Note:**
>
> Do not use the created DB_AUDIT user to connect to the database. This user is created for the sole purpose of maintaining a separate audit schema in which all audit objects and data are stored. You should use the SYSADMIN account to configure the system audit settings using DB Audit GUI.

The first time you enable system audit, DB Audit prompts you to choose a tablespace where DB Audit will create the DB_AUDIT schema objects and store audit trail data. Make sure to pick a tablespace that has sufficient amount of free space.

Although not required, it is highly recommended that you choose a separate tablespace dedicated to storing the audit trail data. By using a separate tablespace, you can segregate the I/O generated by the audit trail from the I/O generated by the application. Space management also becomes easier when the two are separated. If you wish to use a separate tablespace but don't have it yet, cancel the **Select Tablespace** dialog, create a new tablespace, place it on a separate device and then return to DB Audit to enable the system audit option.



You must also perform the following 5 steps:

1. Copy the **dbauditRunner.jar** file to your **[db2 home]/sqllib/function** directory. This file can be found in the DB Audit installation directory.

2. Add the **dbauditRunner.jar** file to the Java CLASSPATH environment variable for the DB2 instance owner and bounce your DB2 instance in order for the new CLASSPATH to take effect. You can also add it to the global system CLASSPATH environment variable.

   Example: *To update environment variables on a Windows system, right-click on the My Computer icon on the Desktop; choose the **Properties** command in the popup menu to display the Properties dialog; then click the **Advanced** tab. On the **Advanced** tab, click either the **Environmental Variables** button or the hyperlink to display the Environmental Variables dialog. In that dialog in the System Variables box, locate the CLASSPATH variable and add the following text (without the quotation marks) to the end of the existing value: "**;***C:\[db2_home]\SQLLIB\FUNCTION\ dbauditRunner.jar***", replacing** *[db2_home] with the actual path. Close all opened dialogs and restart the DB2 instance.*

3. Start the DB Audit interface service on your DB2 server by running the following command in the *[db2 home]/sqllib/function* directory:

   java –jar dbauditRunner.jar

   Note: It is important that the working directory be set to the *[db2 home]/sqllib/function* directory in order for the dbauditRunner.jar process to run correctly.

4. **On Unix systems**: add the *java –jar dbauditRunner.jar* command to the profile of the DB2 instance owner so that it can start automatically when your DB2 computer starts. To have the command start as daemon process in the background, add the ampersand (&) symbol at the end of the command line.

   On Windows systems: use the *d*bauditRunnerSrv.exe program to install the audit process runner service which will start the audit processes automatically when the system starts. This file can be found in the *[DB Audit home]\DB2* directory. To install the service, copy dbauditRunnerSrv.exe to the *[db2 home]\SQLLIB\function* directory and from there run the following command:

   dbauditRunnerSrv.exe /install

5. Schedule periodic runs of Audit Record Flushing and Loading Procedures. See the following topic for detailed instructions on how to do that.

Important Notes: After enabling system-level auditing, you should schedule periodic run of the audit trail loading procedures. These procedures periodically flush audit trail records to the disk and then load the flushed data into the system audit trail tables stored in the database. See the following topic for information on how to schedule the mentioned procedures.

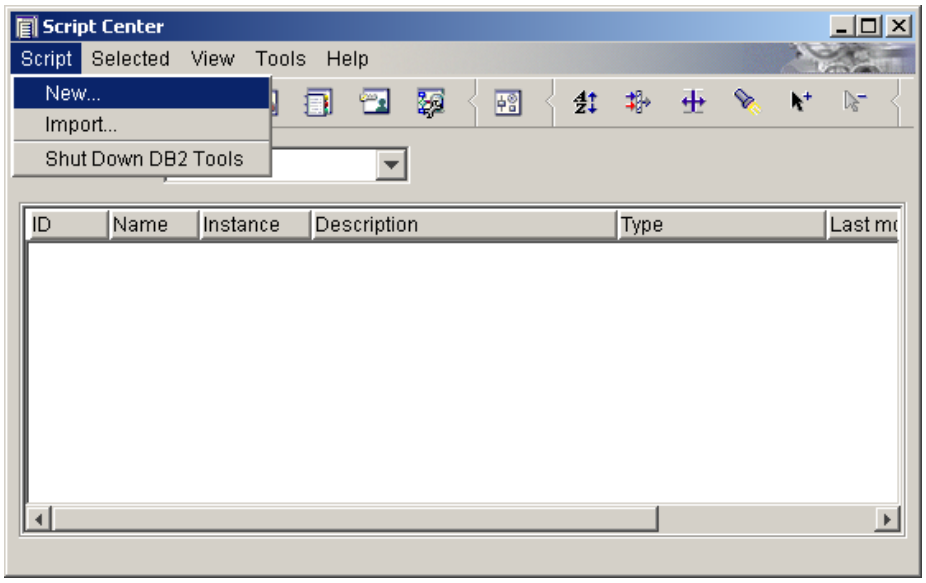
Scheduling System Audit Record Flushing and Loading Procedures.

Due to the nature of auditing, volume of audit log data can grow quite quickly in a short period of time. Depending on the speed of the growing you may need to use different scheduling method for the audit log loading procedures. If the volume is not very high you can use convenient methods available in DB2 Control Center, in particular you can use either Script Center or Task Center tools (whatever is available in your DB2 version).

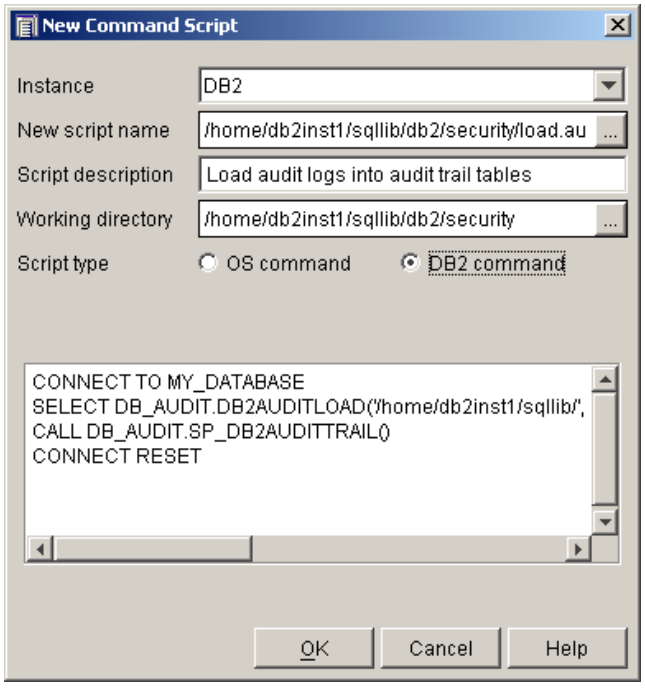
The DB2 Script Center (or Task Center) can be used to schedule runs with 1 hour or more seldom occurrences. If the volume of the data log data is high and the load processing takes a long time (more then 10-15 minutes) or fails with "log full" errors consider using available host operation system methods or standalone schedulers such as 24x7 Scheduler software to schedule more frequent runs of the audit log loading procedures.

The following example demonstrates how to schedule audit log loading procedures using DB2 Script Center.

1. Start DB Script Center. Click **Script/New** menu to create a new job.



2. When the **New Command Script** dialog will appear, fill-in the required job properties. You can choose any location for the script file and also any description for the script. Example values are provided on following screenshot.



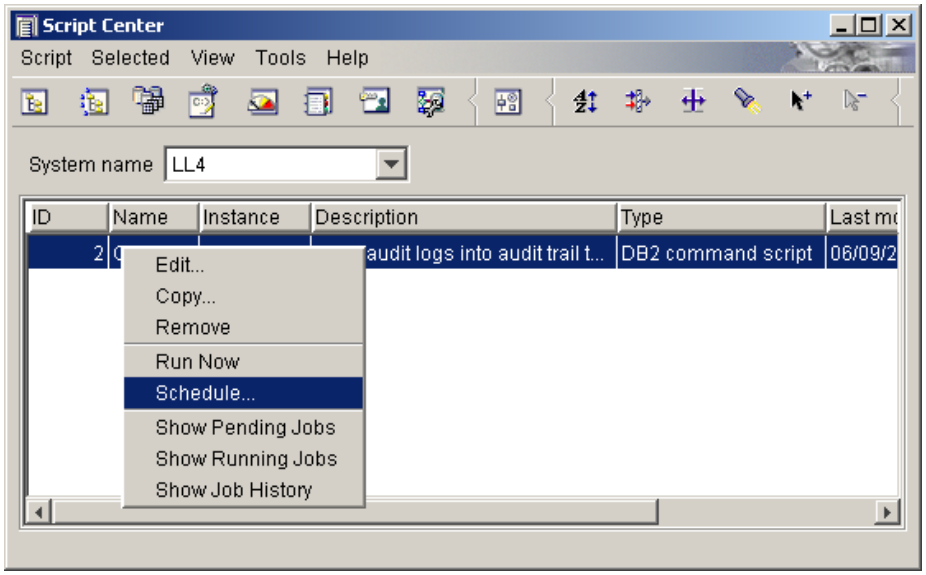
For the command script you must enter the following script

```
CONNECT TO MY_DATABASE
SELECT DB_AUDIT.DB2AUDITLOAD('/home/db2inst1/sqllib/', 'MY_DATABASE' )
       FROM SYSIBM.SYSDUMMY1
CALL DB_AUDIT.SP_DB2AUDITTRAIL()
CONNECT RESET
```

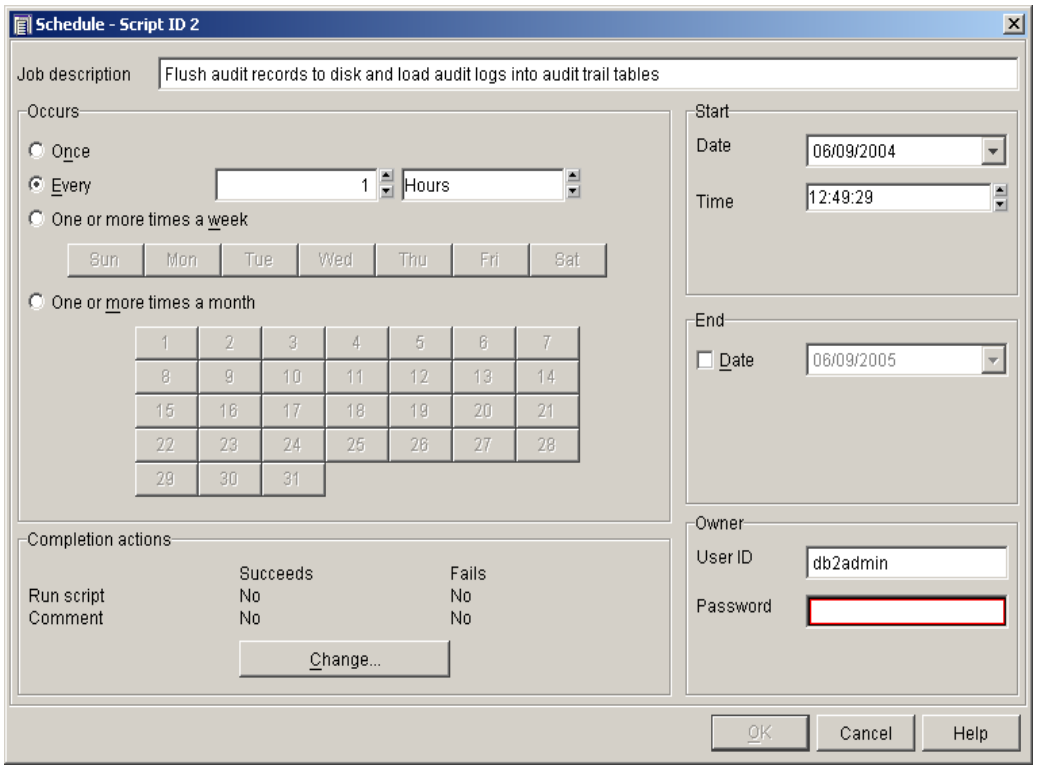
Note: In this script you must replace **MY_DATABASE** with the actual name of your DB2 database. Also replace **/home/db2inst1/sqllib/** with the home directory of your DB2 instance including the **sqllib** directory and the trailing slash characters. For your convenience, the complete script is copied to the clipboard by the DB audit Management Console when you enable the system audit trail facility.

Click the **OK** button to close the script dialog.

3. Now you need to schedule the created script. Right-click on the job line and then select **Schedule** from the popup menu.



This will open the **Schedule Script** dialog.



a) Enter descriptive schedule name, for example, *"Flush audit records to disk and load audit logs into audit trail tables."*

b) Specify the scheduling frequency. We recommend that you run this job at least once per hour, more frequently if your DB2 version supports more frequent occurrences.

c) In the Start box, enter today's date and time. This will be the job's initial start date and time.

d) Enter job owner information. This is the name and password of the user account that will be used to run the job. The account must have sufficient permissions to delete and insert data into DB Audit repository tables.

e) Click the **OK** button to close the Schedule dialog. The job is now scheduled and you can close the Script Center.

DB2: Disabling system audit

1. Click **System Audit/Set Audit Options** command from the DB Audit Expert menu to launch the System Audit Configuration screen.

2. Click the **Disable System Audit** button displayed on the bottom of the System Audit Configuration screen.

3. Use the DB2 Script Center or Task Center to disable or remove system audit loading procedures. For more information on accessing DB2 Script Center, see the previous topic.

DB2: Setting system audit options

Click **System Audit/Set Audit Options** command from the DB Audit Expert menu to launch System Audit Configuration screen.

The System Audit Configuration screen consists of three tab pages:

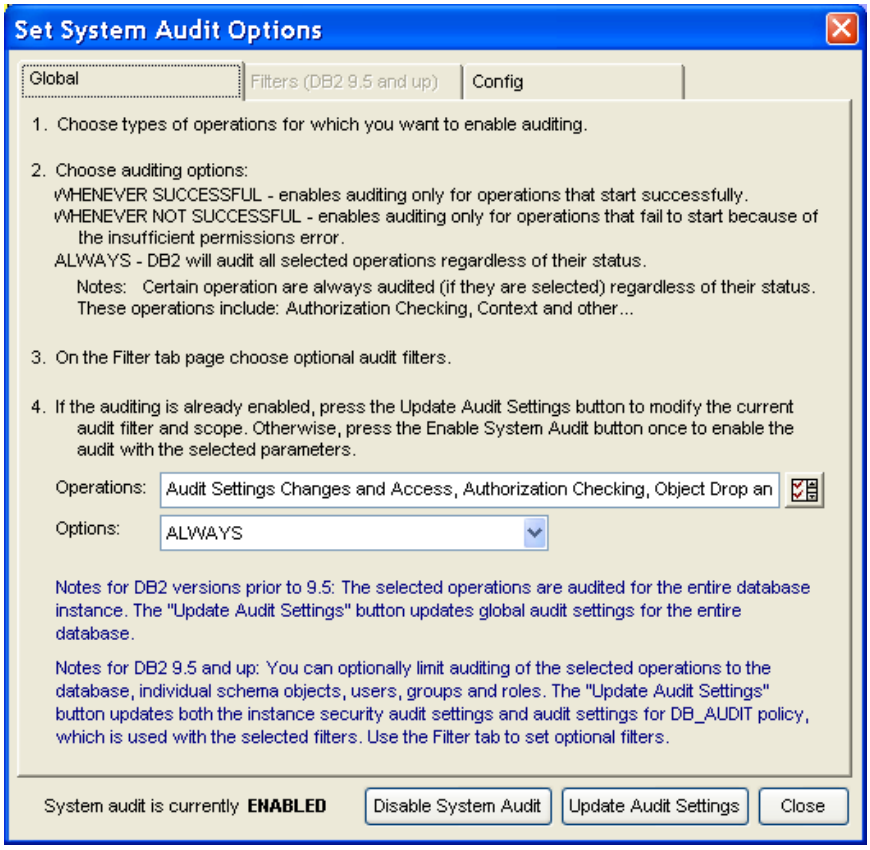
1. **Operations** – use options available on this page to select the categories of database operations to be audited and the conditions under which they will be audited.

2. **Config** – use options available on this page to configure audit trail behavior.

   Tip: Options on this page affect how DB Audit monitors DB2 events and records audit trail data.

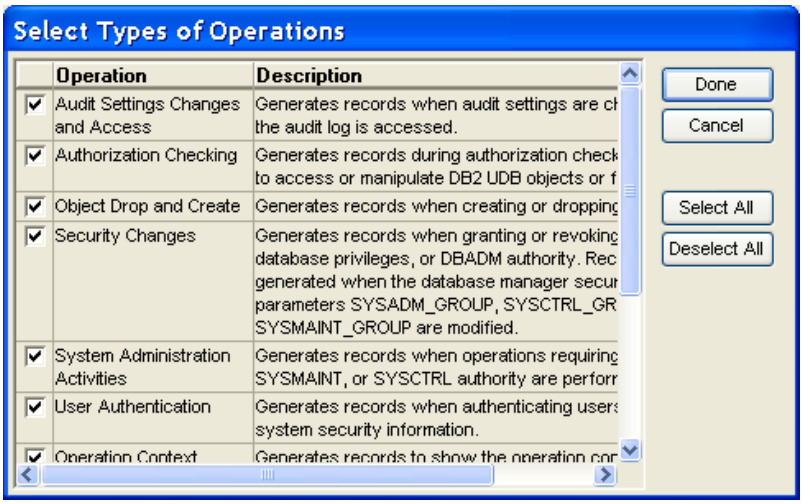
Selecting Operations

The following screenshot demonstrates typical selection for audit Operations settings.



To select which Operations to audit:

1. Activate the **Operations** tab page. Choose types of operations for which you want to enable auditing. Click the **Lookup** button displayed on the **Operations** line to open the **Select Operations** dialog.



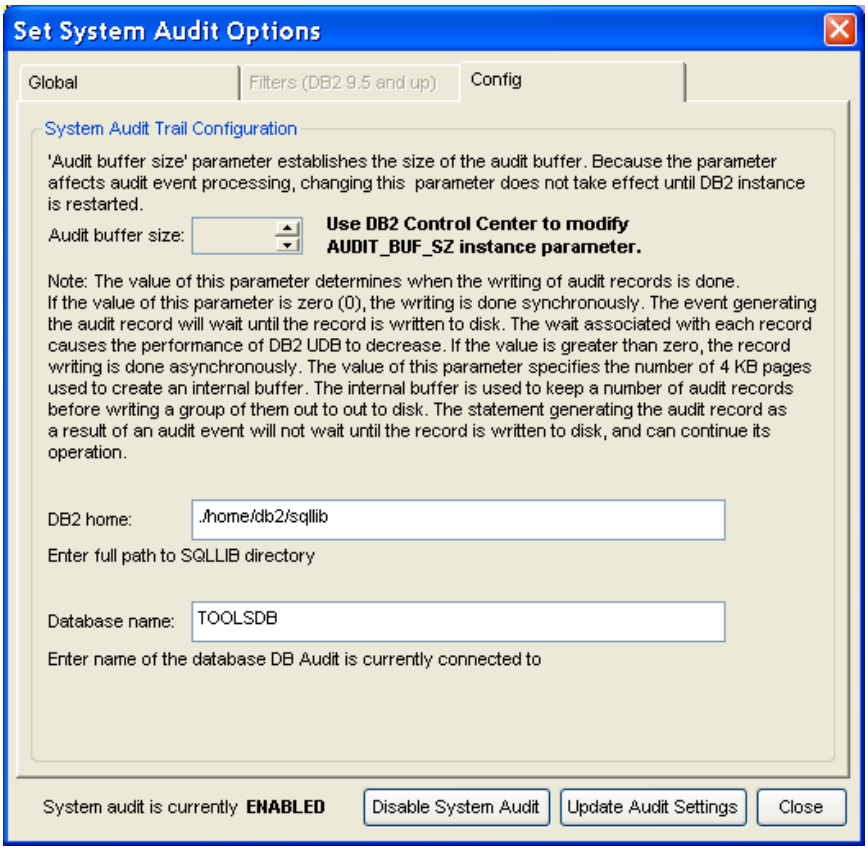
You can select one or more operations by placing a checkmark in the left-most column. To quickly select all listed operations, click the **Select All** button. To quickly deselect all previously selected operations, click the **Deselect All** button. Click the **OK** button to return to the previous screen with all selected operations. To cancel the dialog without making any changes, click the **Cancel** button.

2. Choose the WHENEVER SUCCESSFUL option to enable auditing only for **authorized database operations**. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for failed operations; in other words, attempts to issue **non-authorized commands**. If you select ALWAYS or leave the **Options** blank, DB2 will audit all database operations regardless of success or failure.

3. Click the **Enable System Audit** button or **Update Audit Settings** button (whichever is enabled) to save the new settings and, if necessary, to start up the auditing process. Use the **Update Audit Settings** button if the system audit is already installed and enabled. Use the **Enable System Audit** button if the system audit is not yet installed and you are using the Audit Configuration Settings screen first time. The **Enable System Audit** button performs two actions: it saves the chosen audit settings and installs the DB Audit procedures and catalog tables.

To configure advanced audit options:

1. Activate the **Config** tab page. Specify DB2 home directory on the database server computer.



2. Specify name of the database containing audit trail tables.

MySQL: Configuring System Audit Options

MySQL: Enabling system audit

In order to manage audit options for MySQL, you must connect to your database as ROOT user who also has administrative privileges on the computer running the MySQL instance.

Important Note: In order to enable system audit in MySQL, DB Audit installs several user-defined external functions as well as several tables and procedures in the DB_AUDIT repository database schema. The DB Audit GUI automatically installs all these objects and creates the necessary user and schema when you click the **Enable System Audit** button on the System Audit Configuration screen.

Inf the DB Audit Management Console is running locally on MySQL server computer, it automatically deploys to the server all DLL/SO files required for the auditing services.

If the DB Audit Management Console is running remotely, you must copy the required DLL/SO files manually from the DB Audit Management Console computer to the MySQL server computer. Look for the files in MySQL subfolder within the DB Audit Management Console installation folder.

If you are running MySQL Server versions 5.1.20 or earlier, copy the required DB Audit files to the BIN subfolder of your MySQL instance. If you are running MySQL Server versions 5.1.21 or 5.1.22, copy the required DB Audit files to the LIB subfolder of your MySQL instance. If you are running MySQL Server versions 5.1.23 or later, copy the required DB Audit files to the LIB\PLUGIN subfolder of your MySQL instance.

Make sure to choose the correct versions of required files. The following DLL/SO files are required for different environments and must be copied to their respective systems:

MySQL 32-bit running on 32-bit and 64-bit Windows operation system
- **db_audit_mysql.dll**
- **db_audit_sendmail.dll**
- **24x7ws.dll**

MySQL 64-bit running on Windows operation system, x86-64 architecture
- **db_audit_mysql.dll**

MySQL 32-bit running on 32-bit and 64-bit Linux operation system
- **db_audit_mysql.so**
- **db_audit_sendmail.so**

To turn the auditing on, you must restart MySQL after the audit installation.

Enabling System Audit and Selecting Audit Repository Location

The first time you use DB Audit to configure system audit settings and you click the **Enable System Audit** button, DB Audit automatically installs all required database objects, including several external UDF functions as well as the following audit trail and configuration tables. These tables are used internally to save your audit settings and audit results:

DB_AUDIT.SYS_AUDIT_TRAIL	Used to store audit results (also called system audit trail)
DB_AUDIT.SYS_AUDIT_SETTINGS	Used to store audit settings. Do not modify data in this table directly as this may lead to system audit malfunction.

Both tables are created in the same DB_AUDIT schema.

 **Important Note:**

Do not use the created DB_AUDIT user to connect to your database. This user is created for the sole purpose of maintaining a separate audit schema where all audit objects and data is stored. You should use the ROOT account to configure the system audit settings using the DB Audit GUI.

MySQL: Disabling system audit

1. Click **System Audit/Set Audit Options** command from the DB Audit Expert menu to launch System Audit Configuration screen.

2. Click the **Disable System Audit** button displayed on the bottom of the System Audit Configuration screen.

MySQL: Setting system audit options

Click **System Audit/Set Audit Options** command from the DB Audit Expert menu to launch the System Audit Configuration screen.

The System Audit Configuration screen consists of three tab pages:

1. **Operations** – Options on this page to select which operations will be audited and under what conditions they will be audited.

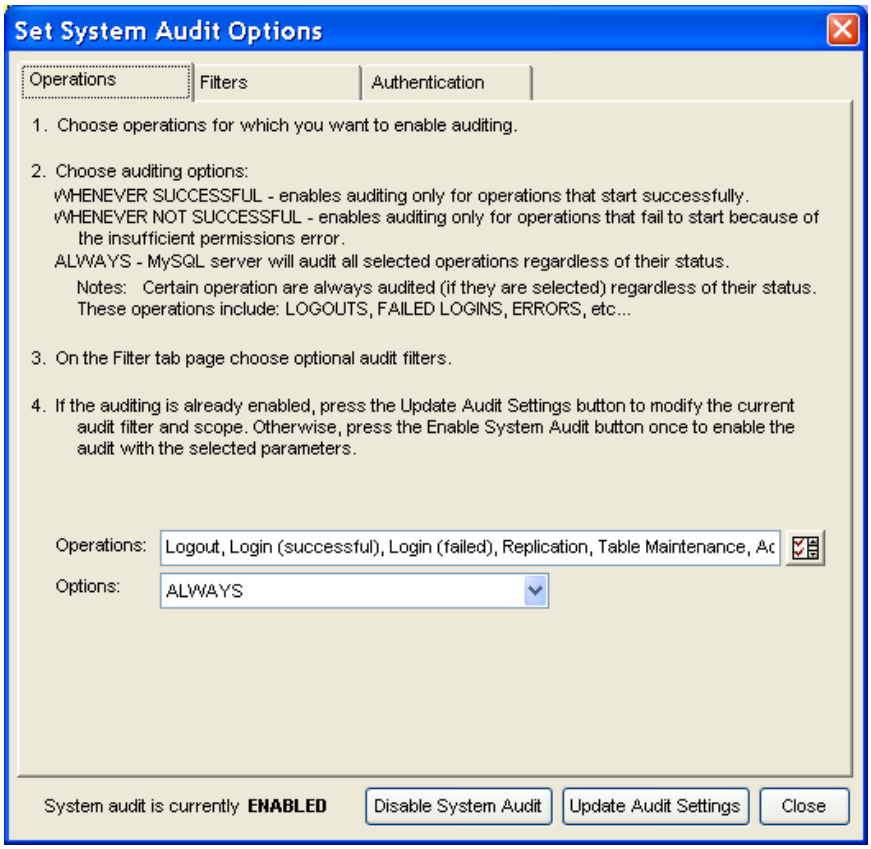
2. **Filters** – Options on this page are used to configure audit operation filters.

    **Tip**: Filters can be applied to the selected audit events to limit the results of the auditing to those events generated by a specific user or application. Filters can also be set to look for specific statements or specific database objects.

3. **Authentication** – Options on this page are used to update the db_audit user password.

Selecting Operations

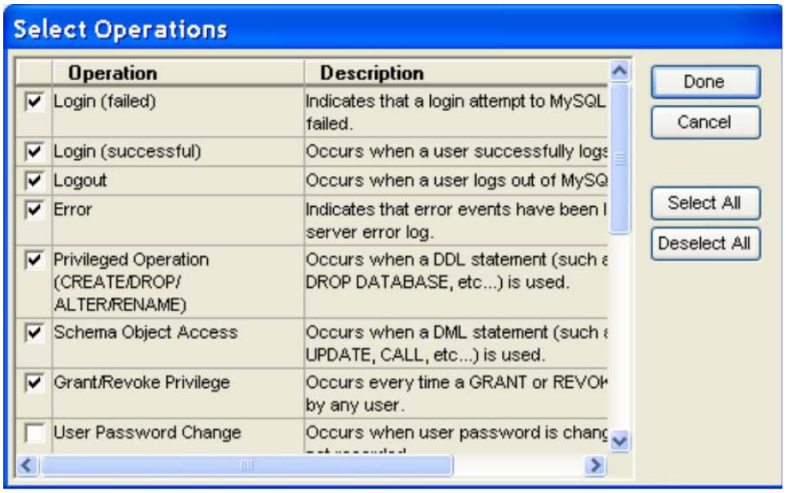
The following screenshot demonstrates typical selection for audit Operations settings.



To select the Operations to audit:

1. Activate the **Operations** tab page. Choose operations for which you want to enable auditing.

   Click the **Lookup** button  displayed on the **Operations** line to open the **Select Operations** dialog.



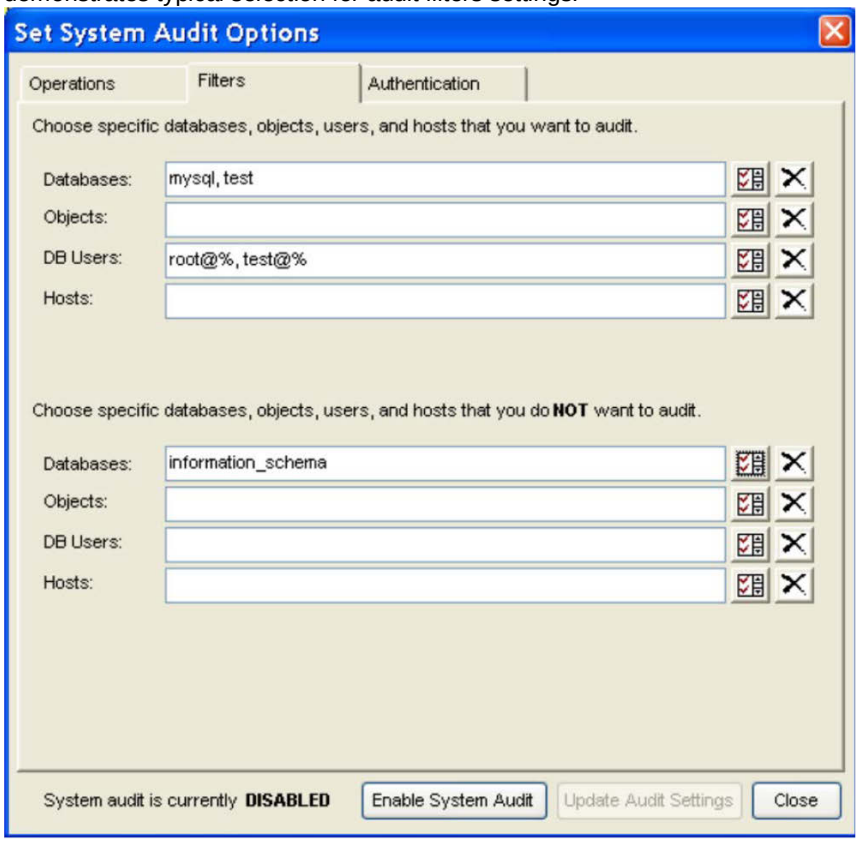
You can select one or more operations by placing a checkmark in the left-most column. To quickly select all listed operations, click the **Select All** button. To quickly deselect all previously selected operations, click the **Deselect All** button. Click the **OK** button to return to the previous screen with all selected operations. To cancel the dialog without making any changes, click the **Cancel** button.

2. Choose the WHENEVER SUCCESSFUL option to enable auditing only for **authorized database operations**. Choose the WHENEVER NOT SUCCESSFUL option to enable auditing only for failed operations; in other words, attempts to issue **non-authorized commands**. If you select ALWAYS or leave the **Options** blank, SQL Server will audit all database operations regardless of success or failure.

3. If you don't want to set up any audit event filters, click the **Enable System Audit** button or **Update Audit Settings** button (whichever is enabled) to save the new settings and, if necessary, to start up the auditing process. Use the **Update Audit Settings** button if the system audit is already installed and enabled. Use the **Enable System Audit** button if the system audit is not yet installed and you are using the audit configuration settings screen for the first time. The **Enable System Audit** button performs two actions: it saves the chosen audit settings and it installs the DB Audit procedures and catalog tables.

To select audit filters:

1. Activate the **Filters** tab page. Choose appropriate audit filters. The following screenshot demonstrates typical selection for audit filters settings.



Multiple filtering parameters can be selected. When combined, they create a single complex filter applied to the audit process. Logical AND operations are used to join different parameters together. For example, if you select *mysql* and *test* databases for the **Databases** parameter and also select *root@%* and *test@%* for the **DB Users** parameter, the audit trail

will contain only records related to user activities for *root* and *test* users in *mysql* and *test* databases.

The following filtering parameters are supported:

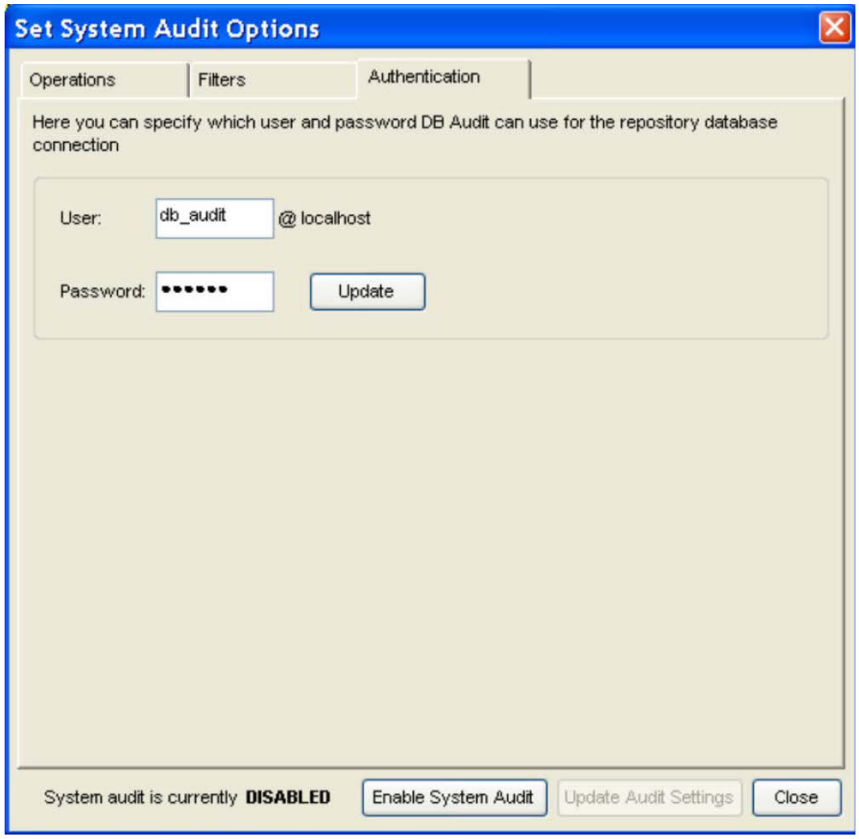
Databases	Use this parameter to limit the auditing to specific databases. Multiple databases can be selected for the filter. Click the **Lookup** button  displayed on the **Databases** line to open the **Lookup Database Names** dialog. The dialog lists all existing databases. Select one or more databases by placing a checkmark in the left-most column. Click the **OK** button to close the lookup dialog and enter the selected databases into the Databases parameter. Click the **Delete** button  on the **Databases** line to clear this parameter.
Objects	Use this parameter to limit the auditing to specific schema objects. Multiple objects from the same schema can be selected for the filter. Click the **Lookup** button  on the **Objects** line to open the **Lookup Object Names** dialog. In the **Lookup Object Names** dialog, select the required schema using the **Schema** drop-down list. This will populate the object names list with all existing objects in the selected schema. Select one or more objects by placing a checkmark in the left-most column. Click the **OK** button to close the Lookup dialog and add the selected objects to the Objects parameter. Click the **Delete** button  on the **Objects** line to clear this parameter.
DB Users	Use this parameter to limit the auditing to specific database users. Multiple users can be selected for the filter. Click the **Lookup** button  on the **DB Users** line to open the **Lookup Database User Names** dialog. This dialog lists all existing database users from all databases. Select one or more users by placing a checkmark in the left-most column. Click the **OK** button to close the lookup dialog and add the selected names to the DB Users parameter. Click the **Delete** button  on the **DB Users** line to clear this parameter.
Hosts	Use this parameter to limit the auditing to specific computers from which users connect to the database. Multiple hosts can be selected for the filter. Click the **Lookup** button  on the **Hosts** line to open the **Lookup Hosts** dialog.

	Important Notes: This dialog lists names of all hosts registered in MySQL settings in the users table. Select one or more hosts by placing a checkmark in the left-most column. Click the **OK** button to close the lookup dialog and add the selected host names to the Hosts parameter. Click the **Delete** button on the **Hosts** line to clear this parameter.

2. If you have already selected operations for which you want to enable auditing, click the **Enable System Audit** button or **Update Audit Settings** button (whichever is enabled) to save the new settings and if necessary startup the auditing process. Use the **Update Audit Settings** button if the system audit is already installed and enabled. Use the **Enable System Audit** button if the system audit is not yet installed and you are using the audit configuration settings screen first time. The **Enable System Audit** button performs two actions: it saves the chosen audit settings and it installs the DB Audit procedures and catalog tables.

To configure security settings for the auditing service user:

3. Activate the **Authentication** tab page. Specify the **user and passwords** parameters. The following screenshot demonstrates typical selection for security settings.



User: It is recommended that you keep the default user account used for the auditing service. You can update the password for this account as required.

 **Important Notes:** You should first change the password for DB_AUDIT user in the database and only then use settings on the **Authentication** tab to update DB Audit settings.

To change the password without restarting the auditing service and updating configuration, use the **Update** button next to the **Password** field.

4. If you have already selected filters and operations for which you want to enable the auditing, click the **Enable System Audit** button or **Update Audit Settings** button (whichever is enabled) to save the new settings and, if necessary, to start up the auditing process. Use the **Update Audit Settings** button if the system audit is already installed and enabled. Use the **Enable System Audit** button if the system audit is not yet installed and you are using the audit configuration settings screen first time. The Enable System Audit button performs two actions: it saves the chosen audit settings and installs the DB Audit procedures and catalog tables.

CHAPTER 4: Data Change Auditing

In addition to System Auditing used to track database and user activity, DB Audit Expert also supports trigger-based data change auditing. Data change auditing must be used in:

- SOX-compliant systems
- Other government regulated systems in which a complete or partial data-change trail must be retained for historical auditing purposes for creating reports on who, when, and what changes were made to the data.

Important Note: Neither System Auditing nor any other network-based network traffic capturing methods allows you to capture and store results of data change operations. Do not be fooled by various so-called non-intrusive auditing methods. While these methods can capture SQL commands sent to the database over the wire, they are unable to capture the results of these commands. Results of any SQL command using bind variables or updating multiple records at once cannot be reliably described from the SQL command text.

For example, consider the following command, *DELETE FROM ACCOUNT_TRANSACTION WHERE trans_date < GetDate() – 5*. The text of this example command cannot be used to determine what data has been touched by the command. Moreover, the impact of this command is time-sensitive and may produce different results if executed several times over a period of time or if executed simultaneously by multiple users. **If your system requires full SOX compliance, you have no choice but to use data change auditing.**

Each audited table requires an audit trigger and a mirror table called "audit trail". DB Audit Expert creates all audit trigger and audit trail tables in a single DB_AUDIT schema. If this schema does not exist, DB Audit Expert automatically creates it for you when you select the first table to be audited.

For each selected table, DB Audit Expert automatically creates both the audit trigger and the mirror table. The mirror table is used as a data audit trail table for storing data changed in the audited table. DB Audit Expert stores one auditing log record for every deleted or newly inserted record and two records for every updated record. In the case of updates, the first log record always stores old values from the updated record while the second log record stores the new values.

How it works

DB Audit Expert enables the tracking of data changes made to a database. DB Audit Expert can record what changes were made and who, when, and from where they were made. Any number of tables or columns may be selected for auditing, and different types of auditing can be specified for different tables. For instance, you can track only inserts on one table and monitor all changes on another table.

As indicated above, the actual auditing of the database is implemented by installing audit triggers on each table selected for auditing and creating audit trail tables to store data from the changed records. The installed data-change auditing is completely transparent to both the existing as well as new applications.

All supported database systems allow multiple triggers defined on a table; hence, even if you already have some existing application triggers, or you add new triggers at a later time, both the application and audit triggers will work simultaneously. The audit triggers will not impact the existing triggers. However, you must check your database settings to ensure that multiple triggers per table and per action are permitted. The order in which triggers are executed is determined by the DBMS and cannot be specified by the user. Most database systems execute triggers in the order these triggers were

created. Refer to the Multiple Triggers Execution Order topic for additional information.

To ensure compatibility with existing applications, DB Audit creates its own database schema in which all repository objects and triggers are created. As a secondary precaution, it always generates unique names for audit triggers and audit trail tables. The algorithm used to generate audit trigger and audit trail table names is different for different database systems. Note that all generated audit objects are preceded by the "AUDIT_" prefix, allowing you to easily distinguish these objects from other database objects and to easily locate them in the database system catalog tables.

ASA, SQL Server: Adaptive Server Anywhere and MS SQL Server do not currently support triggers created in a schema separate from the table tables. In these databases, DB Audit always creates triggers in the audited table schema. This behavior may change in future versions of Adaptive Server Anywhere and MS SQL Server. DB Audit supports flexible trigger schema options and will automatically recognize and locate the correct trigger schema.

In addition to creating audit triggers and tables, DB Audit also automatically creates the following catalog tables. It uses these tables internally to catalog names of generated objects and to save your audit settings:

DB_AUDIT.DATA_AUDIT_TRAIL
DB_AUDIT.DATA_AUDIT_COLUMNS

DB_AUDIT.DATA_AUDIT_USERS
DB_AUDIT.DATA_AUDIT_APPS

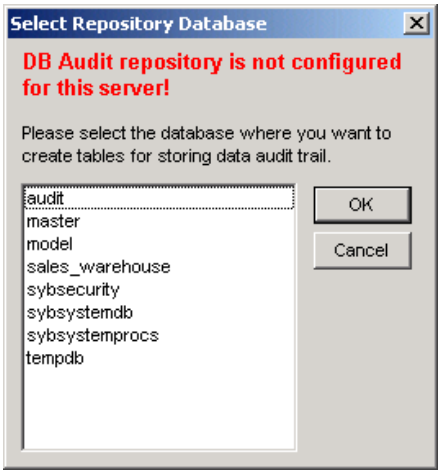
All catalog tables are created in the same DB_AUDIT schema.

To create a DB_AUDIT schema, most database systems require that a user with the same name exists in the database. When you use DB Audit to create the first audit trigger, it automatically creates the required user and, if required, a matching login name.

Important Note:

Do not use the created DB_AUDIT user/login to connect to your database. This user is created for the sole purpose of maintaining a separate audit schema where all audit objects and data is stored. You should use the appropriate administrator account to setup the audit triggers using DB Audit GUI.

SQL Server, ASE: When you create the first audit trigger, DB Audit prompts you to choose a database where DB Audit will create the DB_AUDIT schema objects and store audit trail data.



Although not required, it is highly recommended that you choose a separate database dedicated to storing the audit trail data. Using a separate database greatly simplifies overall system maintenance and reduces system and storage overhead requirements for audit data. In addition, the production database is also separated from the audit data and thus performance and space management is simplified. Audit data can also be secured easily. If you wish to use a separate database but have not yet created one, cancel the **Select Repository Database** dialog, create a new database and place it on a separate device. Then return to DB Audit and repeat the trigger generation process.

Important Notes:

- Make sure the database selected for the repository has the "Select into/bulk copy" option enabled. If it does not, either use available database graphical management tools (such as SQL Server Enterprise Manager or Sybase Central) or execute **sp_dboptions** system procedure manually to enable the "Select into/bulk copy" option. To enable this option using **sp_dboptions** system procedure, run the following SQL command, replacing *'repository db name'* parameter with the real repository database name.

```
exec sp_dboptions 'repository db name', 'select into/bulkcopy',
'on'
go
```

- If the repository database differs from the database containing the audited table, make sure database users who change data in the audited tables also exist in the repository database. To add a user to the repository database, you can use available database graphical management tools (such as SQL Server Enterprise Manager or Sybase Central) or execute **sp_adduser** system procedure. To add a user using **sp_adduser** system procedure, run the following SQL commands, replacing *'repository db name'* parameter with the real repository database name and replacing *'login'* and *'user'* parameters with the real login name and database user name.

-
```
use 'repository db name'
go
exec sp_adduser 'login', 'user'
go
```

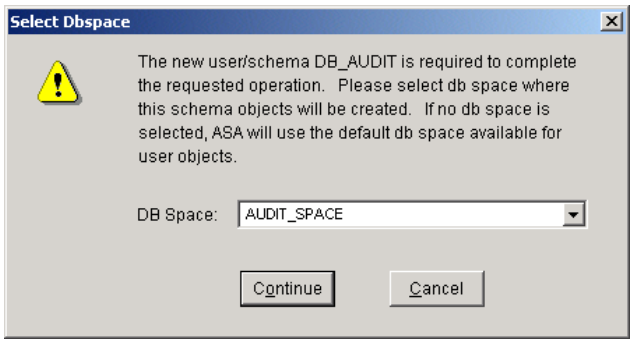
-

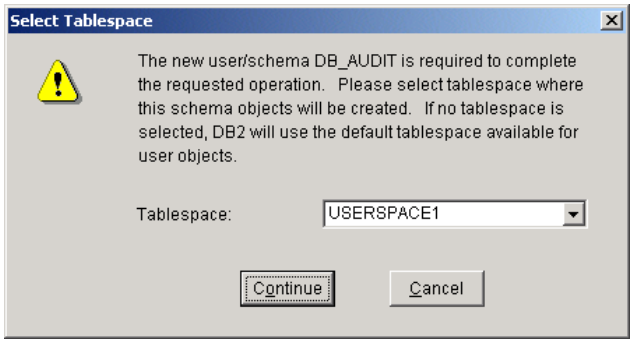
Oracle: When you create the first audit trigger, DB Audit prompts you to choose a tablespace where DB Audit can create the DB_AUDIT schema objects and store audit trail data. Choose a tablespace that has a sufficient amount of free space, but never use the SYSTEM tablespace.

Although not required, it is highly recommended that you choose a separate tablespace dedicated to storing audit trail data. Using a separate tablespace allows you to segregate I/O generated by the audit trail from I/O generated by the application. Space management also becomes easier when the two are separated. If you wish to use a separate tablespace but have not yet created one, cancel the **Select Tablespace** dialog, create a new tablespace, and place it on a separate device. Then return to DB Audit to repeat the trigger generation process.

ASA: When you create the first audit trigger, DB Audit will prompt you to choose a dbspace where DB Audit can create the DB_AUDIT schema objects and store audit trail data.

If you wish to use a separate dbspace but have not yet created one, cancel the **Select DBspace** dialog, create a new space and place it on a separate device, then return to the DB Audit and repeat the trigger generation process.





DB2: When you create the first audit trigger, DB Audit will prompt you to choose a tablespace where DB Audit will create the DB_AUDIT schema objects and store audit trail data. Make sure to pick a tablespace that has a sufficient amount of free space.

Although not required, it is highly recommended that you choose a separate tablespace dedicated to storing the audit trail data. Using a separate tablespace allows you to segregate I/O generated by the audit trail from I/O generated by the application. Space management also becomes easier when the two are separated. If you wish to use a separate tablespace but haven't yet created one, cancel the **Select Tablespace** dialog, create a new tablespace, and place it on a separate device. Then return to DB Audit to repeat the trigger generation process.

DBMS privileges required

The table below describes default privileges that DB Audit Expert grants to the DB_AUDIT user when creating DB_AUDIT schema and audit triggers. The table also describes privileges you may need to grant to users who have access to the DB Audit Report Viewer application.

Oracle 7 and 8	**Oracle 9 and 10** requires explicit grants given by SYS	**DB2**	**Sybase ASA**	**Sybase ASE** and **MS SQL Server**

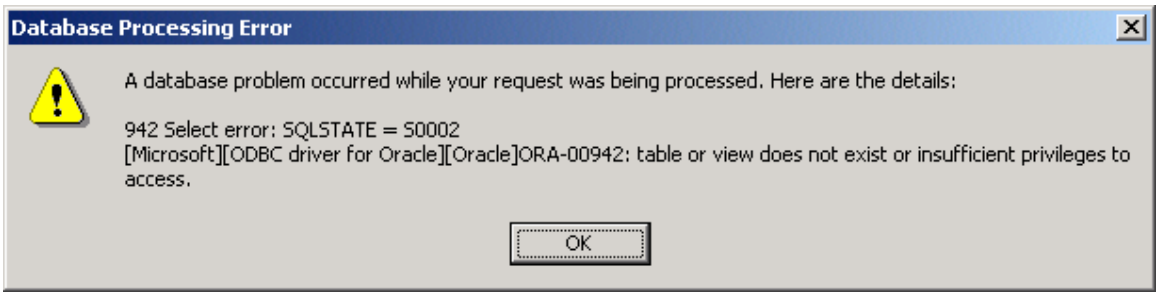
`SELECT ANY TABLE`
 	
`EXECUTE ANY PROCEDURE`
	`SELECT ON sys.v_$session`
	
	`SELECT ON sys.v_$mystat`
	
	`EXECUTE ON sys.dbms_lob`
	
	`EXECUTE ON sys.dbms_output`
	
	`EXECUTE ON sys.dbms_job`
	
	`EXECUTE ON sys.dbms_util`
	
	`SELECT, DELETE ON sys.audit$`
		`CREATETAB ON DATABASE`	`RESOURCE`	<none>
				
				It might be necessary to give certain permissions to audited application users who make changes in the audited tables. For example, if a trigger is set up with the email alert option, users would need execute permissions for the following system procedures:
				
				`xp_startmail`
				`xp_sendmail`
				`xp_stopmail`
				
				DB Audit by default grants INSERT permission to PUBLIC for all audit trail tables. If the audit repository is created in another database, all audited application users must be created in the chosen repository database.

Privileges required for Report Viewer users include:

```
GRANT SELECT on db_audit.data_audit_trail to 'user'

GRANT SELECT on db_audit.data_audit_columns to 'user'
```

If you get the following error message, it means you don't have sufficient privileges to access DB Audit repository tables. Contact your database administrator to obtain the required privileges. The actual message text may vary in different database systems.



Guidelines

1. The data audit trail tables contain the same columns as their associated audited tables. In addition, they also contain several additional columns for storing change tracking information, such as the name of the database user who changed the data, the network id of that user, the terminal from where the operation was initiated, the application used, the type of transaction, and the timestamp indicating the exact date and time of the change.

2. If the table being audited contains a unique column, the data from the data audit trail can be used to recover the audited table in case of data loss or just to rollback the changes to any given point in time.

3. DB Audit Expert supports multiple triggers on multiple selected tables. You can perform data change auditing on virtually unlimited number of tables simultaneously.

4. The volume of audit trail information collected during data-change auditing can grow very quickly for various transaction tables. Hence, you should regularly archive audit records and purge the audit trail. Once you have collected the required information, archive the audit records of interest and purge the audit trail of this information. DB Audit provides all the tools necessary to automate the archiving and purging processes. See Data-change audit trail management topic for more information on how to set up these processes.

5. Performance, space and management overheads go hand in hand with auditing. These considerations can dictate the level of detail and time span when this data is collected and retained online. Use prudence when enabling data-change auditing and base this on firm business requirements.

Limitations

Direct Data Load and Table Truncation

Data modifications that do not fire table triggers cannot be audited, including:

- TRUNCATE TABLE operations
- Direct data loads using DBMS native utilities with the fire-trigger options turned off. Examples are Bulk Copy (BCP) operations without the appropriate options to execute triggers in Microsoft and Sybase database servers, SQL*Loader in Oracle, DB2 LOAD command and so on.

Large Character/Binary Data Support (BLOB)

 **SQL Server:** Microsoft SQL Server versions 7, 2000, 2005 and 2008 do not support access to *text*, *ntext*, and *image* data type columns using standard INSERT INTO…SELECT operations. For SQL Server version 6.5, DB Audit supports auditing and recording of BLOB columns. In a SQL Server 7 database in 6.5 compatibility mode, BLOB data types are allowed in triggers. However, they return NULL instead of the actual value when referenced. When running in Version 7 compatibility mode, you cannot reference text columns in a trigger at all. There is no work around for this in a trigger. DB Audit must omit *text*, *ntext*, and *image* data type columns in generated audit triggers.

 **Oracle:**

- **Oracle versions 8.0 and later** do not support referencing *long* and *long raw* data type columns in table triggers. As a result of this, *long* and *long raw* columns cannot be audited. There is no work around for this in a trigger. DB Audit must omit *long* and *long raw* data type columns in generated audit triggers.
- **Oracle versions 7.x** allow *long* and *long raw* data types in triggers. However, they return NULL instead of the actual value when referenced if the value cannot be converted to a constrained data type (such as *char* and *varchar2*). If the data can be converted to a constrained data type, it can be saved in the audit trail. The data however, will be truncated to include only the first 32,000 bytes.
- When Oracle OCI functions or the DBMS_LOB package is used to update LOB values or LOB attributes of object columns, triggers defined on the table containing the columns or the attributes will not be executed. As a result, such data-changes cannot be audited.

Object Columns and Nested Tables

 **Oracle:** Performing DML operations directly on nested table columns does not cause Oracle to execute triggers defined on the table containing the nested table column. If you want to audit such changes, you should define direct triggers on nested tables. To find out names of nested tables, you can use the following SQL query:

```
SELECT table_name
FROM dba_nested_tables
WHERE owner = '<your_schema_name>' AND
       parent_table_name = '<your_table_name>' ;
```

User Tracking

Oracle: DB Audit captures complete user tracking information including user network name, database user id, the name of the terminal from which the connection was made, and the application/process name. Because of certain Oracle internal limitations, the terminal and process name information could be truncated to the first 80 characters.

SQL Server: DB Audit captures the following user tracking information: user login name, database user id, the name of the terminal from which the connection was made and the application name (if available). If SQL Server is configured to use database-side authentication, the captured login name is the user's SQL Server login name; otherwise, it is the user's Windows login name.

Some applications use a single SQL Server login to make database connections. Often the login is hard-coded within the application code. In such situations, try to use the captured terminal name to track down real users.

In some cases, the client software through which the database connection is made, rather than the SQL Server itself, provides the terminal names and application names. Not all types of the client software set these values automatically. The recent types and versions of client software available from Microsoft, such as *ADO* and *MDAC,* do this automatically, but some older versions of *DBLib* do not. If this is not set automatically, the client application itself is responsible for setting these values in the database connection. If you do not see values in the data audit trail log reports for *Application Name* or *Terminal*, this is because the client application has not provided those values.

ASE: DB Audit captures the following user tracking information: user login name, database user id, the name of the terminal from where the connection was made, and the application name (if available). If ASE is configured to use database-side authentication, the captured login name is the user's ASE login name. Otherwise, it is the user's Windows login name.

Some applications use a single ASE login to make database connections. Often the login is hard-coded within the application code. In such situations try to use the captured terminal name to track down real users.

In some cases, the client software through which the database connection is made, rather than ASE itself provides the terminal names and application names. If you do not see values in the data audit trail log reports for *Application Name* or *Terminal*, this is because the client application has not provided those values.

ASA: DB Audit captures the following user tracking information: database user name. Other user/connection identification information is not available in ASA

DB2: DB Audit captures the following user tracking information: database user id (*current sqlid*) and database login name. This is the same as the user network name because DB2 always uses host system authentication. Other user/connection identification information is not available in DB2.

There are limits on what triggers can accomplish. As of DB2 v6, you cannot define triggers on:

- a system catalog table
- PLAN_TABLE
- STATEMENT_TABLE
- DSN_FUNCTION_TABLE
- Any table with a three-part name

Multiple Triggers Execution Order

The DB Audit installed data-change auditing is completely transparent to both existing and new applications. All supported database systems allow multiple triggers defined on a table. Hence, if you already have some existing application triggers or create new triggers, both the application and audit triggers will work simultaneously. However, the order in which these triggers are executed is determined by the DBMS and cannot be explicitly specified. Most database systems execute triggers in the order these triggers were created. Therefore the following scenarios are possible:

1. An audit trigger is executed first. The trigger records changes in the table and optionally sends an email alert. An application trigger is subsequently executed in the same transaction. The trigger validates changes, and a business rule rejects the changes by causing the transaction to roll back.

   In this scenario the audit records will be also removed (rolled back) from the audit trail table, but the email alert that was already sent cannot be reversed or recalled.

2. An audit trigger is executed first. The trigger records changes in the table and optionally sends an email alert. An application trigger is subsequently executed in the same transaction. The trigger validates the changes and business rules, and corrects certain new values before these values are saved to the table.

   In this scenario the audit records will store the changed data as the user entered it before the application trigger subsequently modified the data.

Enabling Data Change Audit

Enabling auditing of any table or a group of tables is as simple as 1, 2, 3. Simply perform the following three steps:

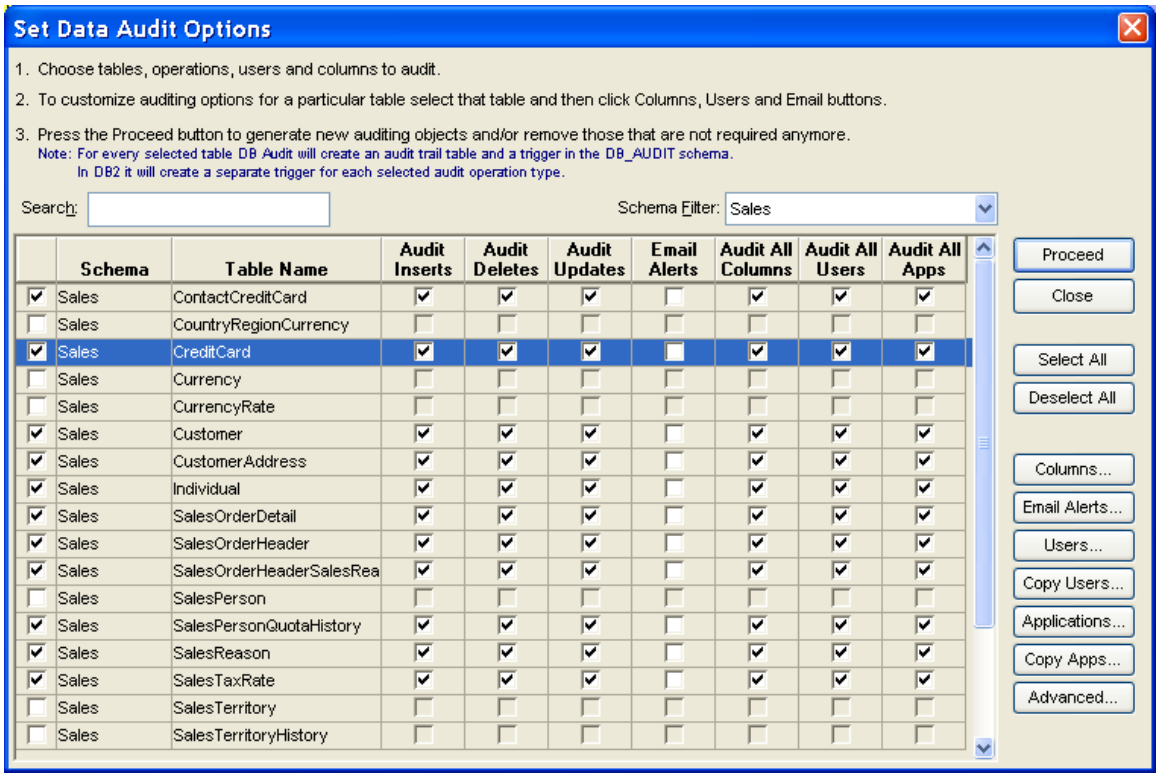
1.  **SQL Server and ASE:** Use the Database drop-down box displayed on the DB Audit top Toolbar (refer to DB Audit Main GUI Controls topic for details) to choose the database storing the tables to be audited.
    **Oracle, ASA, MySQL, and DB2:** Skip this step.

2. Click the **Data Audit/Set Data Audit Options** menu to open the **Set Data Audit Options** dialog. Tables that already have DB Audit triggers set up will appear with a check mark in the left-most column, as well as with the proper selection of operations and columns.

   If the table list is very long, use the **Schema** drop-down box to filter tables by owner. This displays only tables for the chosen owner. To quickly locate a particular table, start typing the name of that table in the **Search** edit field. DB Audit will search and highlight the first table

whose name matches the entered text.

3. Check the checkboxes in the left-most column for tables you want to enable audit on, or use the **Select All** button to enable all displayed tables in one keystroke.



Click the **Proceed** button to set up new triggers, modify existing triggers, or remove unnecessary triggers.

Note:
When you press the **Proceed** button, DB Audit compares selected table, operation and column settings with the existing database triggers. DB Audit automatically removes triggers and audit trail tables for tables that were deselected. It also generates new triggers and audit trail tables for newly selected tables, and rebuilds triggers and audit trail tables for tables whose selection of audit operations or columns were modified. After completing the Set Data Audit Options operation, DB Audit automatically reconciles the current selection against database system catalog tables and deselects tables for which audit triggers were not created successfully.

Choosing Audit Scope

In a busy database system, complete data-change auditing can lead to many gigabytes of audit data collected over short periods of time. DB Audit supports several types of audit filters for fine-tuning the audit process. Such filters can be instrumental in improving database performance and reducing space requirements for audit trail tables.

Please do not confuse audit filters with audit report filters. Report filters are used to reduce volume of information printed by audit reports to meaningful summaries and focused details. They are applied during the report generation process, while audit filters are applied during data change auditing

process. In other words, audit filters are used to control amount of the information collected and recorded in the audit trail tables.

The following audit controls and filters are available.

- Audit all or only specific database schemas and tables.
- For selected tables, audit all types of data change operations or choose only specific types of SQL operations such as INSERT, DELETE, or UPDATE.
- For selected tables and operations, audit changes in all columns or audit changes only in specified columns.
- For selected tables and operations, capture all values in every modified record or capture values only in specified columns.
- For selected tables and operations, audit changes made by any user or only those made by specified users.
- For selected tables and operations, audit changes made by any application or only those made by specified applications.

 You can select any combination of filtering options listed above. All of the mentioned controls are accessible directly from the Set Data Audit Options dialog. The remainder of this chapter provides information about each of the supported audit filter types.

Selecting Auditable Operations

The three checkboxes to the right of the table name (in the **Audit Inserts**, **Audit Updates**, and **Audit Deletes** columns) represent types of SQL operations you may wish to audit for the selected tables. By default, all three are checked; however, you can choose any combination you like. You can also choose different operations for different tables.

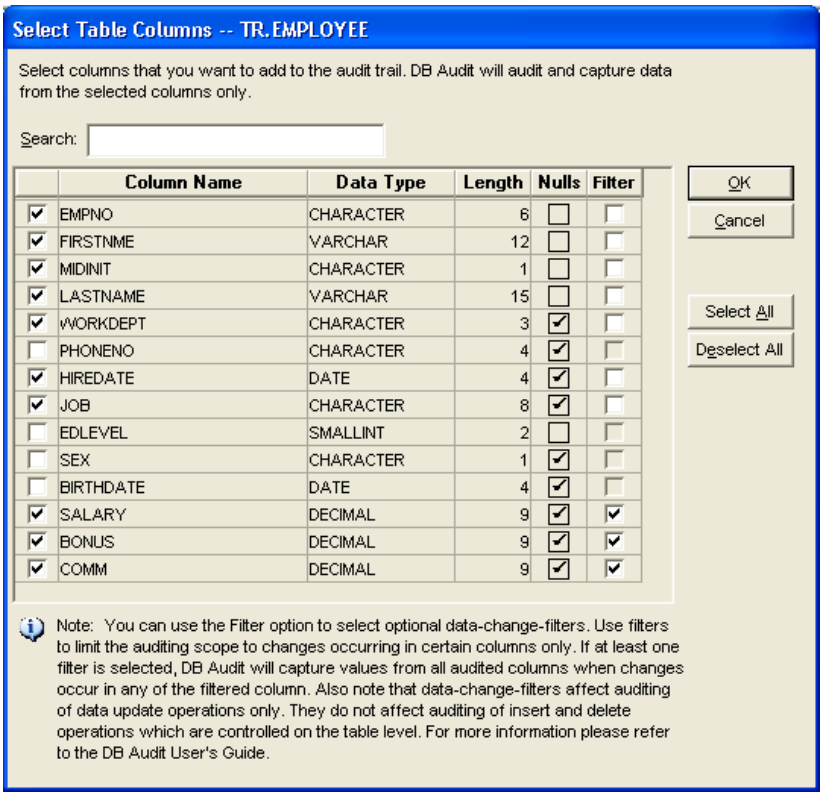
 Regardless of the selected operations, DB Audit always creates only one trigger per table (except for DB2 as explained in the next paragraph). The trigger is executed only in events that match the selected operations.

 **DB2:** DB2 currently does not support multiple-event triggers. Because of this, DB Audit creates a separate trigger for each selected operation. Thus, if all DELETE, INSERT and UPDATE operations are selected, DB Audit creates three separate triggers.

Selecting Audit Trail Columns and Auditable Changes

The check mark in **Audit All Columns** column indicates whether or not all columns are selected for auditing. This checkbox is checked by default, indicating that the data from all columns will be captured and saved in the data audit trail whenever an audit trigger is executed for a particular table. By all means, you should audit every column that you have reason to audit. However, you should carefully consider which columns might not necessarily have to be audited. Bear in mind that the more audited columns you have, the more data will accumulate in the audit trail tables and the more the overall performance and storage requirements of the database will be affected by the need to read/write and store this audit data.

To choose which columns must be audited select a particular table and click the **Columns** button. The **Select Columns** dialog will appear. You will see all columns for the currently selected table. Choose which table columns will be audited by checking the checkboxes in the left-most column. Click the **OK** button to accept changes and close the dialog, otherwise click the **Cancel** button to leave the current column selection unchanged.



You can use the Filter option to select optional data-change-filters. Use filters to limit the auditing scope only to changes occurring in certain columns. If at least one filter is selected, DB Audit will capture values from all audited columns when changes occur in any of the filtered column. Also note that data-change-filters affect auditing of data update operations only. They do not affect auditing of INSERT and DELETE operations, which are controlled on the table level. For more information, see Selecting Auditable Operations topic in this chapter.

Setting User-level and Application-level Audit Filters

You can optionally specify which users and/or applications you do or do not want to audit. By default all users and applications are audited. To apply user-level and/or application level filters, use the **Users** and the **Applications** button. To quickly copy filter definitions to other selected tables, use the **Copy Users** and the **Copy Applications** buttons.

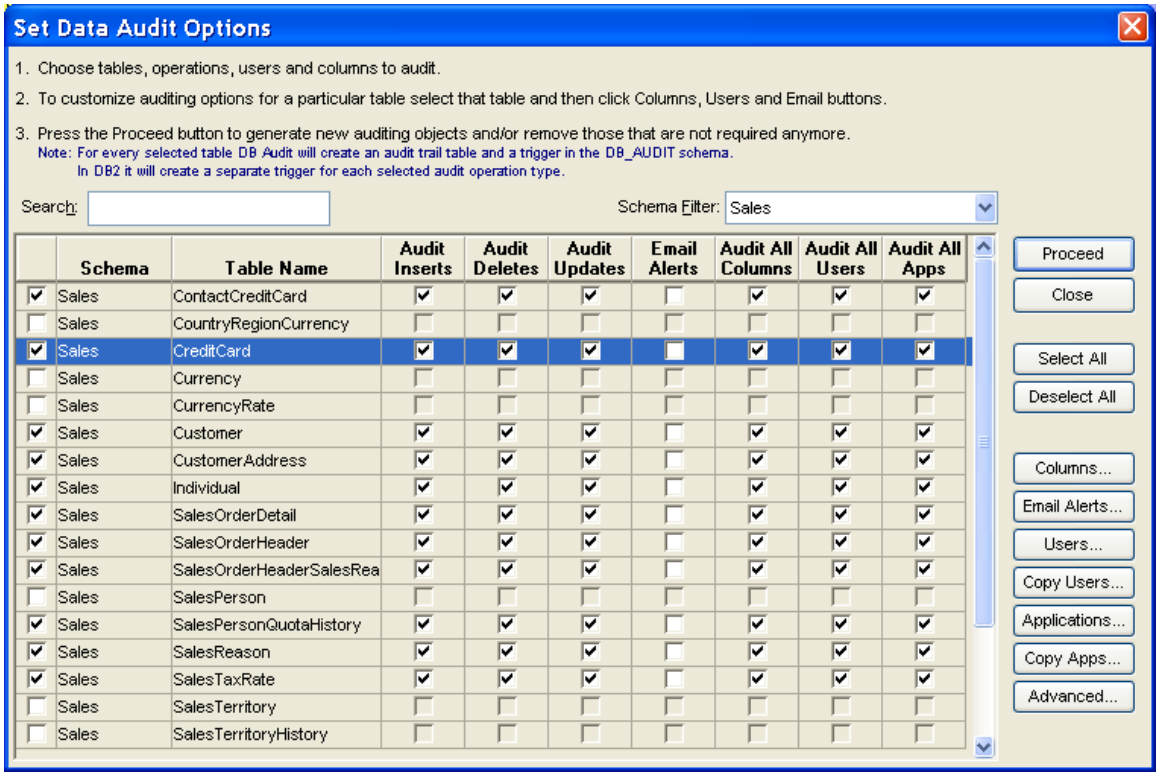
Both user-level audit filters and application-level audit filters can be configured during the audit trigger generation process. The Set Data Audit Options screen contains controls that can be used to configure the audit filters. For more information on the trigger generation process, see Enabling Data Change Audit topic in this chapter.

Notes:

- Different audit filters can be used with different audited tables, if audit filters are defined as

table-specific.

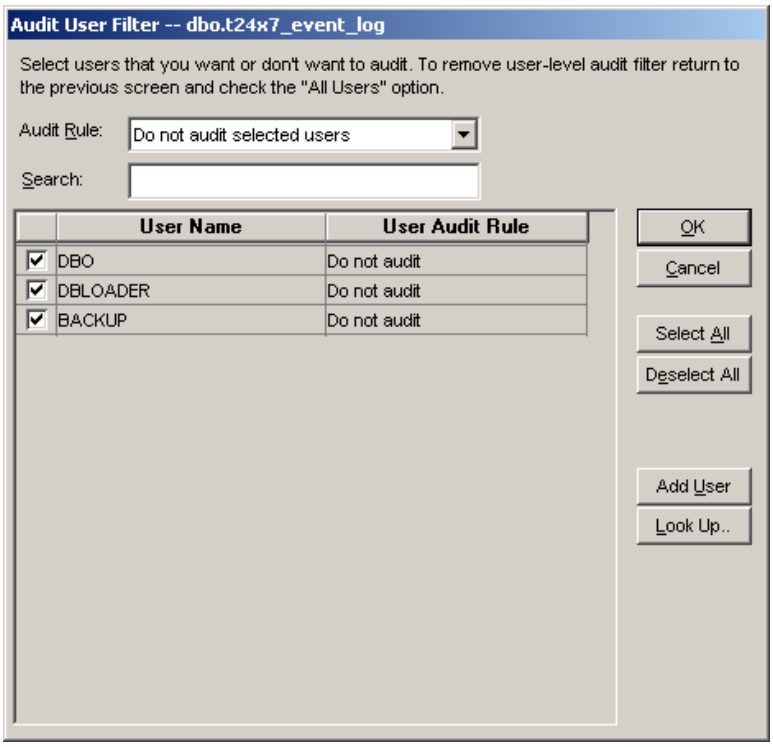
- User-level filters have higher priority over application-level filters. For example, assume that you choose never to audit user BACKUP and always audit the ISQL application. If you then connect to ISQL as BACKUP user and make changes in the audited table, your changes will not be captured and recorded in the audit trail report. For this reason in the audit report you will not see that user BACKUP made changes in the database.



User-level filters:

In the screenshot shown above, the **Audit All Users** column (second from right) indicates whether or not all users are selected for auditing. By default, this checkbox is checked, indicating that changes by any user will be captured and recorded in the data audit trail when an audit trigger is executed for a particular table. You should audit every user that you have reason to audit, but bear in mind that the more users you choose to audit, the more data will accumulate in the audit trail tables, and the more the overall performance and storage requirements of the database will be affected by the need to read/write and store this audit data.

To choose which users must be audited, select a particular table and click the **Users** button. The **Audit User Filter** dialog will appear. The dialog displays all users that have been previously selected for audit filtering for any table. You can use the **Add User** button to add additional user names that haven't been entered before. Choose which users will be added to the table-level user filter by checking the checkboxes in the left-most column.



To quickly add one or more users, you can use the Look Up User option. To open the Lookup User Names screen, click the **Look Up** button. This will display a list of all users currently defined in the database. Choose which users will be added to the table-level user filter by checking the checkboxes in the left-most column. Click the **OK** button to close the dialog and add selected users, or click the **Cancel** button to return to the previous screen without making any changes.

Use the **Audit Rule** drop-down list, to choose whether or not you want to audit selected users.

Note: You can choose to apply different rules to different tables, but you cannot choose to apply different rules to different users within the same table. Within a table, however, you can choose to exempt specified users from auditing for that table.

Click the **OK** button to accept changes and close the dialog, otherwise click the **Cancel** button to leave the current user selection unchanged.

To remove a previously configured user-filter, check the **Audit All Users** checkbox on the Set Data Audit Options dialog. DB Audit will clear the user-level filter.

Note: To ensure that user names are not case sensitive, DB Audit internally stores all user names converted to upper case. This does not affect its ability to audit users whose names in the database are in lower case.

To quickly copy user-level filters to other tables, use the **Copy Users** button on the **Set Data Audit Options** dialog. This will copy user-level filters from the highlighted table to all other tables currently displayed on the screen and selected for auditing. If the highlighted table has no user-level filters then DB Audit will clear any user-level filters that might be already set for other selected tables. You can use the **Schema Filter** drop-down box to change the table display filter. To copy the same user-level filter to all tables and schema in the database first select [All non-system tables] option for the **Schema Filter**.

Application-level filters:

The right most column, **Audit All Apps,** indicates whether or not all applications are selected for auditing. By default, this checkbox is checked indicating that changes by any application will be captured and recorded in the data audit trail whenever an audit trigger is executed for a particular table. By all means, you should audit every application that you have reason to audit. However, bear in mind that the more applications you have chosen for auditing, the more data will accumulate in the audit trail tables, and the more the overall performance and storage requirements of the database will be affected by the need to read/write and store this audit data.

 **Note:**

From DB Audit point of view, the application name is whatever your database reports as a name or client application or process connected to the database. Different database systems use different methods to identify application process names as explained below.

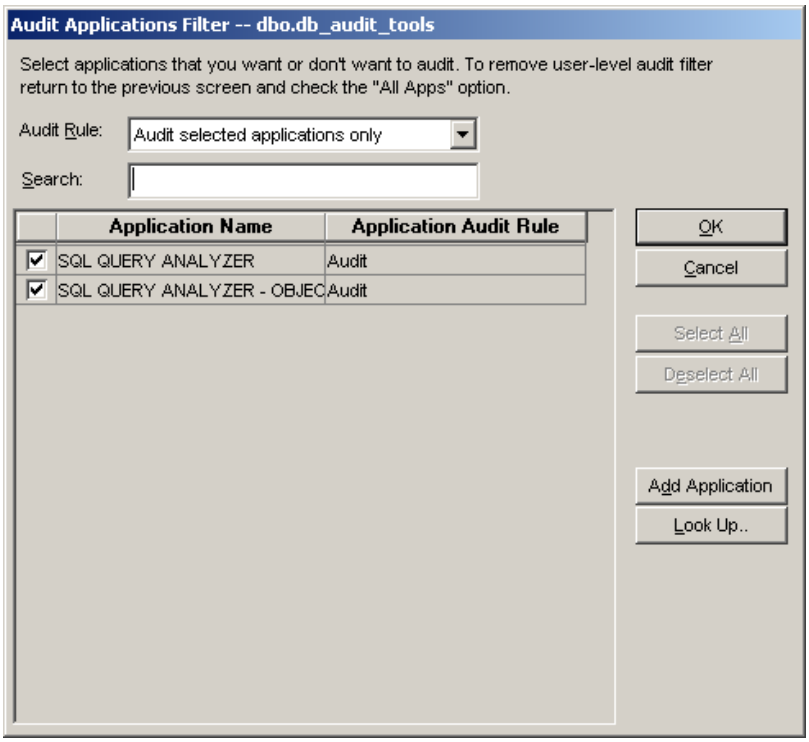
 **Oracle:** In Oracle, the application name is reported as the name of the operation system process. The process name is operation system specific; for example, SQL*Plus for Windows is reported as SQLPLUSW.EXE.

 **ASE, ASA, SQL Server:** In Sybase and Microsoft SQL Server databases, application name is supplied by the client application as free text and can be NULL if none is supplied. In recent versions of SQL Server, the operating system's process name is used for the application name if none is supplied by the actual application.

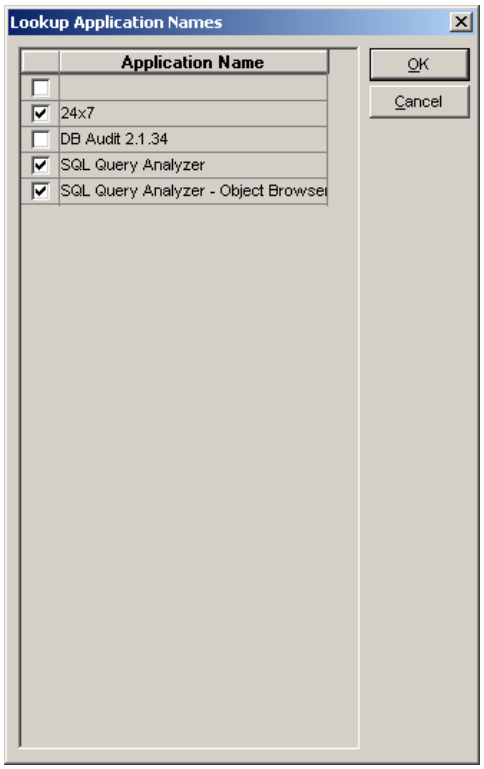
 **DB2:** DB2 does not currently report names of connected applications and processes in a human readable format, and therefore the application-filters cannot be used with DB2 at this time.

 **MySQL:** MySQL does not currently report names of connected application and processes, and therefore the application-filters cannot be used with MySQL at this time.

To choose which applications must be audited, select a particular table and click the **Applications** button. The **Audit Application Filter** dialog will appear. The dialog displays all applications that have been previously selected for audit filtering. You can use the **Add Application** button to add additional application names. Choose which applications will be added to the table-level application filter by checking the checkboxes in the left-most column.



To quickly add one or more applications that are currently connected to the database, you can use the Look Up Application option. To open the **Lookup Application Names** screen, click the **Look Up** button. This will display a list of all applications currently setup in the database.



Choose which applications will be added to the table-level application filter by checking the checkboxes in the left-most column. Click the **OK** button to close the dialog and add selected applications, or click the **Cancel** button to return to the previous screen without making any changes.

Note:

The **Lookup Application Names** screen only lists currently connected applications. If you don't see an application name you want to add, simply cancel the dialog click the **Add Application** button and then enter the name of the application. You must enter the name exactly as it is reported by the database.

Using items available in the **Audit Rule** drop-down list, choose the appropriate audit rule. In other words, choose whether or not you want to audit selected applications.

Note: You can choose to apply different rules to different tables, but you cannot choose to apply different rules to different users within the same table. Within a table, however, you can choose to exempt specified users from auditing for that table.

Click the **OK** button to accept changes and close the dialog; otherwise, click the **Cancel** button to leave the current application selection unchanged.

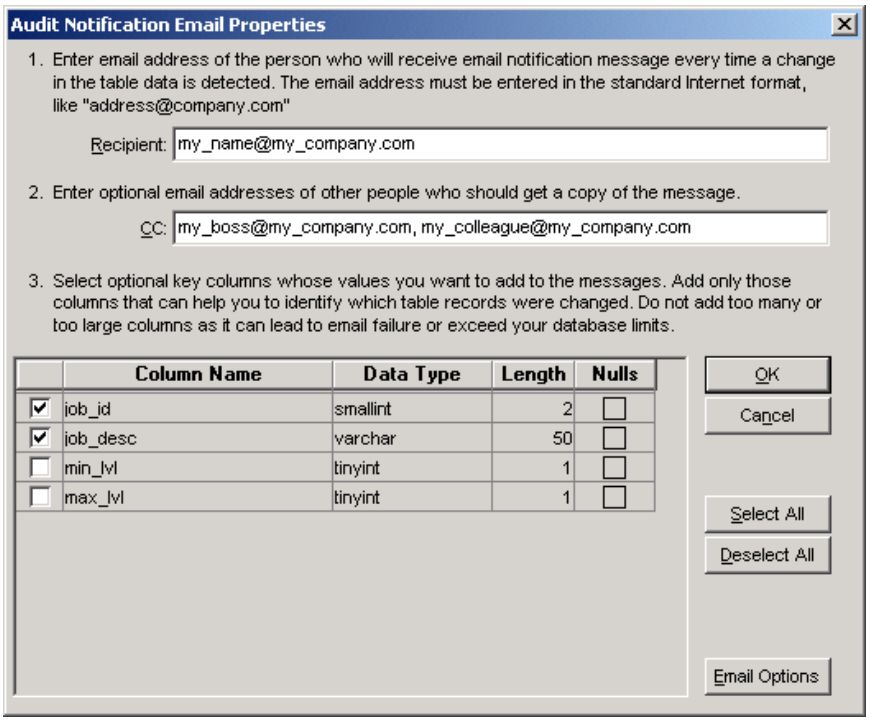
To remove a previously configured application-filter, check the Audit All Applications checkbox on the **Set Data Audit Options** dialog. DB Audit will clear the application-level filter.

Note: To ensure that user names are not case sensitive, DB Audit internally stores all user names converted to upper case. This does not affect its ability to audit users whose names in the database are in lower case.

To quickly copy an application-level filter to other tables, use the **Copy Applications** button on the **Set Data Audit Options** dialog. This will copy the application-level filter from the highlighted table to all other tables currently displayed on the screen and selected for auditing. If the highlighted table has no application-level filter, then DB Audit will clear any application-level filters that might be already set for other selected tables. You can use the **Schema Filter** drop-down box to change the table display filter. To copy the same application-level filter to all tables and schema in the database, first select the [All non-system tables] option for the **Schema Filter**.

Setting Email Alerts

If you wish to add automatic email alerts that are sent to you whenever a change in a particular table occurs in the selected events, click the **Email** button. The **Audit Notification Email Properties** dialog will appear.



Enter properties values as instructed on the screen and click the **OK** button to accept the changes and close the dialog; otherwise click the **Cancel** button to leave the current settings unchanged.

If you have not yet set up the DB Audit email procedure, use the **Email Options** button to set it up. For more information, refer to the Configuring Email Setting for Data Change Alerts topic.

If everything is configured properly, you should receive automatic email messages whenever a change is detected. The email messages that arrive in your email program's Inbox will have the string '*DB Audit Change Notification*' as the subject and a message text indicating who made the change, when and from where. Below is an example message:

Change Time: 28-JUN-02 10:15:22
User NT232\JackSmith just made changes in table HR.EMPLOYEES
Change Type: RECORD DELETED
Record Key(s): EMPLOYEE_ID: 100, FIRST_NAME: Steven, LAST_NAME: King

Configuring Email Setting for Data Change Alerts

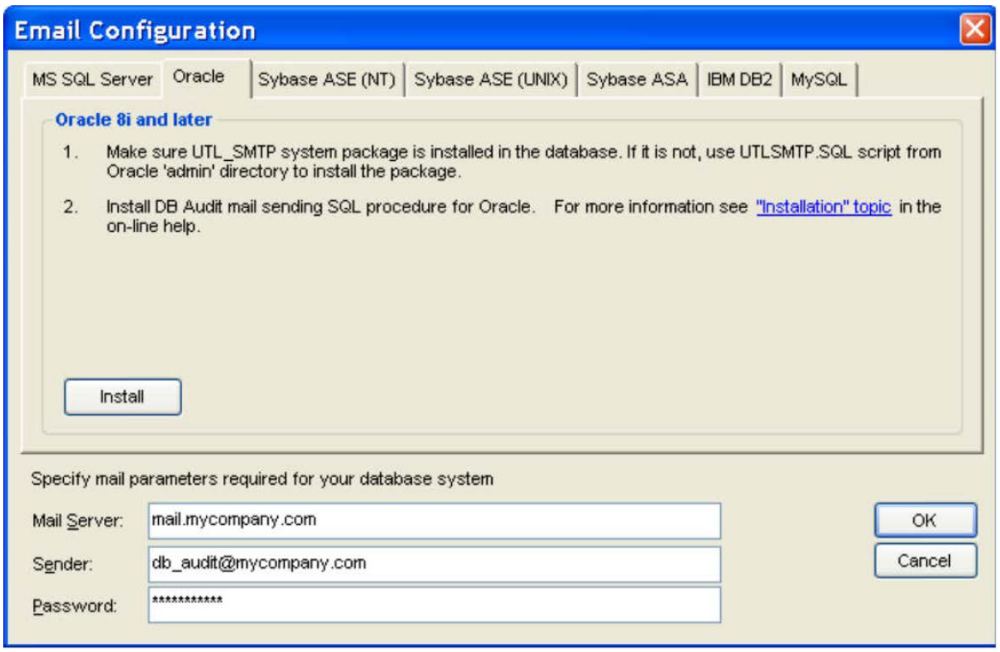
DB Audit supports the ability to generate real-time email alerts to key personnel when changes occur to sensitive data. The alert handlers are installed as a part of the data-change audit trigger generation process, but the actual procedure that sends the email messages must be installed separately.

DB Audit sends change notification alerts using *DB_AUDIT.SP_AUDIT_SENDMAIL* procedure. This procedure is invoked from alert handlers generated within data-change audit triggers whenever trigger code is executed by DBMS.

To install the email sending procedure:

1. Click **Tools/Email Configuration** menu or click the **Email** button on the Toolbar. The Email Configuration screen will appear.

2. Follow instructions on the screen to prepare and install the procedure that can send emails. Click the **Install** button to install the procedure in the database when all other requirements are met.



3. Specify your email server name and login parameters as outlined below:

   If your DBMS uses SMTP interface for email processing:
   Mail Server – Enter your email server network name or IP address.
   Sender – Enter a default email address that can be used to login to the email server and send outgoing emails.
   Password – If required for authentication on your email server, enter your password

   If your DBMS uses MAPI interface for email processing:
   Mail Server – Enter a valid MAPI profile name. The profile name must exist on the database server computer.
   Sender – Enter a default email name that can be used to login to the MAPI.
   Password – If required for the authentication, enter your password.

4. Click the **OK** button to save new settings. The new email settings are saved in the repository as a part of the database profile information. If you delete the profile, your settings will be lost. Because settings are saved as a part of database profile data, you can specify different parameters for different database systems.

   Note: The parameters for the email server, sender and password will not affect the existing data audit triggers. DB Audit will use these parameters the next time you create new triggers or update any existing triggers that have email alerts option enabled.

Note:

- Not all database systems support email functions. Please refer to Feature matrix by DBMS topic for additional information.

- **ASE, SQL Server:** When changes are made in a database table being set up for auditing, Sybase ASE and MS SQL Server execute audit triggers using the security credentials of the user who made the changes rather than the credentials of the user who created the audit trigger. As a result, the *SP_AUDIT_SENDMAIL* procedure is also executed using the credentials of the user

who made the changes. You must grant execute permissions for this procedure to every user who can make changes in the audited tables. Failure to grant execute permissions may lead to trigger errors and user inability to make data changes in the database.

To grant the execute permission you can use the following Transact-SQL command:

```
GRANT EXECUTE ON DB_AUDIT.SP_AUDIT_SENDMAIL to user
go
```

In the command shown above replace *'user'* parameter with the real database user name.

Setting User Name Mapping

In most cases when tracking data changes in the database, DB Audit is able to obtain the operating system user name and terminal name of the user being audited. This information is extracted from the database connection parameters or session data. However, in some cases this may not be possible. For example, this can happen when users first connect to a central application server or a web-based service and where the application server maps all users to a single database user id (as a sort of a super-user). The application server then connects to the database as that super-user. As a result, DB Audit and the database see these connections as concurrent sessions established by one super-user. Because all sessions are appear with the same operating system and database user names, it is usually no possible to trace back to individual users and distinguish who made what changes in the database tables.

To aid in auditing of such applications, you can configure DB Audit to call a user-defined stored procedure passes database session id and other values as input parameters and then returns the unmapped individual user name as its output parameter. This way you can create your own extensions to the DB Audit Expert audit processing.

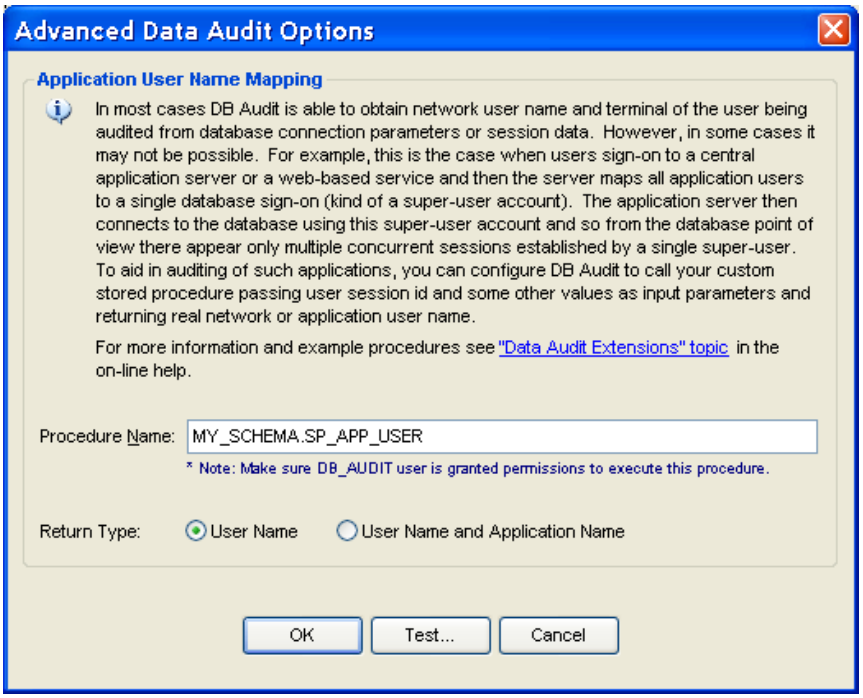
 **DB2:** Not all versions of DB2 support calling user defined stored procedures from database triggers. Check your DB2 manual to find out if your database supports CALL statement in SQL programs.

Using a user-defined procedure for user name mapping

To specify which user-defined stored procedure must be called in the event of data changes:

1. Open the **Set Data Audit Options** dialog. See "Selecting Tables, Columns, and Operations to Audit" topic in this chapter for instructions on opening this dialog.

2. Click the **Advanced** button displayed on the right hand side of the dialog. The **Advanced Data Audit Options** dialog will open.



3. In the **Procedure Name** field, type the full name of the user-defined stored procedure including the schema name.

   SQL Server, ASE, MySQL: If the procedure resides in a database different from the database containing the audited tables, you must enter the fully qualified procedure name starting with the database name.

4. If the procedure uses only one output parameter for the user name, select the User Name option. iI it returns both a user name and application name or module name, select the **User Name and Application Name** option.

5. To test your procedure, click the **Test** button; otherwise, click the **OK** button to save changes and close the dialog.

   Note:
   The specified procedure will be used in new triggers generated on new tables and in existing triggers when they are rebuilt. Similarly, changing a previously defined procedure name only affects new triggers that will be created or modified after the change. Existing triggers will not be affected until you rebuild them. To rebuild existing triggers immediately after you enter or change user-defined procedure name, set the Schema Filter on the Set Data Audit Options screen to [Show all tables] and then click the Proceed button.

Parameter specification for the user name mapping stored procedure

The procedure must accept three input parameters and return a single output parameter. The procedure name and parameters names can be anything. Parameters are passed to the procedure by parameter position and not by name. The following table describes the required parameter specification.

Position	Data Type	Description
1	varchar(30)	**SQL Server, ASE, MySQL:** This parameter is used to pass the name of the database where the audit event occurred. **Oracle, ASA, and DB2:** NULL value is always passed through this parameter. You must define this parameter in the procedure specification, but you can ignore it inside the procedure code.
2	integer	**SQL Server, ASE and ASA:** This parameter is used to pass system process id @@**spid** of the user session being audited. **Oracle:** This parameter is used to pass database session id **SID** of the user session being audited. **DB2:** This parameter is always 0 because DB2 presently doesn't support numeric database session identifiers. **MySQL:** This parameter is used to pass connection the **thread id** of the user session being audited.
3	varchar(80)	This parameter is used to pass the name of the database application being audited. This parameter can optionally be defined as an output parameter. **DB2:** In versions prior to v8, NULL value is always passed through this parameter because application name is not available in DB2 environment. You must define this parameter in the procedure specification, but you can ignore it inside the procedure code. In versions v8 and later, the value of **client applname** special register is used for the application name. **MySQL:** NULL value is always passed through this parameter because application name is not currently available in MySQL environment. You must define this parameter in the procedure specification, but you can ignore it inside the procedure code. If you know the application name, you can define this parameter as an output parameter and use it to provide the application name to DB Audit.
4	varchar(30)	This is the output parameter the stored procedure must use to return the result.

Oracle, DB2, ASE and ASA: The user name mapping procedures can be created either as a native SQL procedure or as a Java procedure, if your database supports Java procedures. When deciding which language to use, you should carefully consider potential performance issues and perform the necessary stress testing.

Important Notes:

- Make sure the DB_AUDIT user gets explicit permissions to execute the specified procedure. To grant this permission run the following command in your favorite database tool:

  GRANT EXECUTE ON [your procedure] TO DB_AUDIT;
- Make sure your procedure returns some kind of [Unknown] value when it is unable to map user name to the session ID.

- Failures in your procedure may prevent users from making changes in your application database tables. Perform comprehensive testing of your procedure before using it for the auditing.

Implementation tips for mapping user names

- If your application server maintains a central application-level audit table or a user connectivity log or statistics table, you can code your custom stored procedure to access that resource and return a valid user name from there. If you are not sure which table is used for connectivity logging, request this information from your application vendor.
- If you are running a web based application that does not take user names but instead logs IP addresses of web site visitors, you can use visitor IP addresses instead of user names for tracking and auditing purposes.
- If you have control over application code and can modify it or can request that the application vendor modify it for you, consider modifying the application code to write the user name, logon time and session id to a log table in the database where it can be accessed by your stored procedure. You can also contact SoftTree Technologies Consulting Services if you need help making these changes.
- If your application logs user information to operating system files rather than database tables, consider creating a file monitoring process at the system level to watch for changes in the log file and immediately load incremental changes into a corresponding database table that can be accessed by your custom stored procedure. Contact SoftTree Technologies Consulting Services if you need help with developing a comprehensive file monitoring and loading solution.
- If is a good idea to periodically purge old data from the user connection log table in order to keep table size to a minimum and to ensure that your procedure does not become a bottleneck in the auditing process. If your DBMS permits, consider pinning the user connection log table to the database cache so that the table data is always loaded in memory and can be accessed quickly without requiring any disk I/O.

Example user-name mapping procedures

SQL Server, ASE and ASA

```
CREATE PROCEDURE user_proc(@db_name varchar(30),
                           @spid integer,
                           @app_name varchar(80),
                            @user_name varchar(30) OUTPUT)
AS
BEGIN
   -- if application name is like Web% return user name from the log
   -- table, otherwise return current session user name
   IF @app_name NOT LIKE 'Web%'
      SELECT @user_name = suser_sname()
   ELSE
      SELECT @user_name = isNull(max(user_name), '[Unknown user]')
      FROM app_log_table
      WHERE spid = @spid
        AND logon_time = ( SELECT max(logon_time)
                           FROM app_log_table
                           WHERE spid = @spid
                             AND logon_time >= DateAdd(d, -1, GetDate())
                         );
END
```

Oracle

```
CREATE PROCEDURE user_proc(db_name varchar2,
                           spid integer,
                           app_name varchar2,
                           user_name OUT varchar2)
IS
BEGIN
   -- if application name is like Web% return user name from the log
   -- table, otherwise return value of the user system variable
   IF app_name NOT LIKE 'Web%' THEN
      user_name := user;
   ELSE
      SELECT nvl(max(t.user_name), '[Unknown user]')
      INTO user_name
      FROM app_log_table t
      WHERE t.sid = spid
        AND t.logon_time = ( SELECT max(t2.logon_time)
                             FROM app_log_table t2
                             WHERE t.sid = spid
                               AND logon_time >= sysdate - 1
                           );
   END IF;
END;
```

DB2 (sample procedure with 1 output parameter)

```
CREATE PROCEDURE user_proc(IN db_name varchar(30),
                           IN spid integer,
                           IN app_name varchar(80),
                           OUT user_name varchar(30))
SPECIFIC DB2ADMIN.user_proc
LANGUAGE SQL
P1: BEGIN
   -- if application name is like Web% return user name from the log
   -- table, otherwise return value of the user system variable
   IF app_name NOT LIKE 'Web%' THEN
      SET user_name = current sqlid;
   ELSE
      SELECT coalesce(max(t.user_name), '[Unknown user]')
      INTO user_name
      FROM app_log_table AS t
      WHERE t.sid = application_id()
        AND t.logon_time =( SELECT max(t2.logon_time)
                            FROM app_log_table AS t2
                            WHERE t2.sid = application_id()
                              AND t2.logon_time >= current timestamp - 1
                          );
   END IF;
END P1
```

DB2 (sample procedure with 2 output parameter)

```
CREATE PROCEDURE user_proc(IN db_name varchar(30),
                           IN spid integer,
                           OUT app_name varchar(80),
                           OUT user_name varchar(30))
SPECIFIC DB2ADMIN.user_proc
LANGUAGE SQL
```

```
P1: BEGIN
   -- if application name is like Web% return user name from the log
   -- table, otherwise return value of the user system variable
   IF app_name NOT LIKE 'Web%' THEN
      SET user_name = current sqlid;
   ELSE
      SELECT coalesce(max(t.user_name), '[Unknown user]'), 'SalesModule'
      INTO user_name, app_name
      FROM app_log_table AS t
      WHERE t.sid = application_id()
        AND t.logon_time =( SELECT max(t2.logon_time)
                            FROM app_log_table AS t2
                            WHERE t2.sid = application_id()
                              AND t2.logon_time >= current timestamp - 1
                           );
   END IF;
END P1
```

MySQL

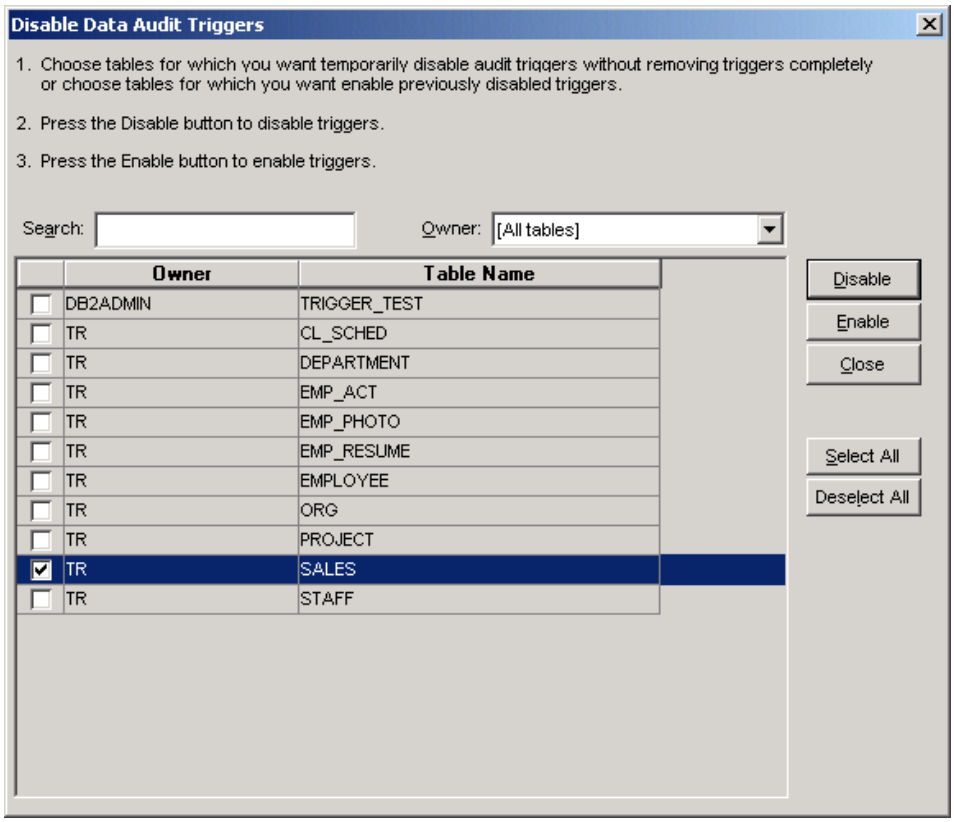
```
CREATE PROCEDURE user_proc(db_name varchar(30),
                           spid integer,
                           app_name varchar(80),
                           OUT user_name varchar(30))
BEGIN
   -- if application name is like Web% return user name from the log
   -- table, otherwise return value of the user system variable
   IF app_name NOT LIKE 'Web%' THEN
      SET user_name = current_user;
   ELSE
      SELECT coalesce(max(t.user_name), '[Unknown user]')
      INTO user_name
      FROM app_log_table AS t
      WHERE t.sid = spid
        AND t.logon_time =( SELECT max(t2.logon_time)
                            FROM app_log_table AS t2
                            WHERE t2.sid = spid
                              AND t2.logon_time >= current_timestamp - 1
                           );
   END IF;
END
```

Disabling Auditing Temporarily

Once audit triggers have been created on a table, you can selectively enable or disable those triggers without destroying the audit defining for the table.

To enable or disable triggers:

1. Click the **Data Audit/Enable/Disable Triggers** menu. The **Disable Data Audit Triggers** dialog will appear



2. Check the triggers that you wish to disable and then click the **Disable** button. Check the triggers that you want to enable and then click the **Enable** button.

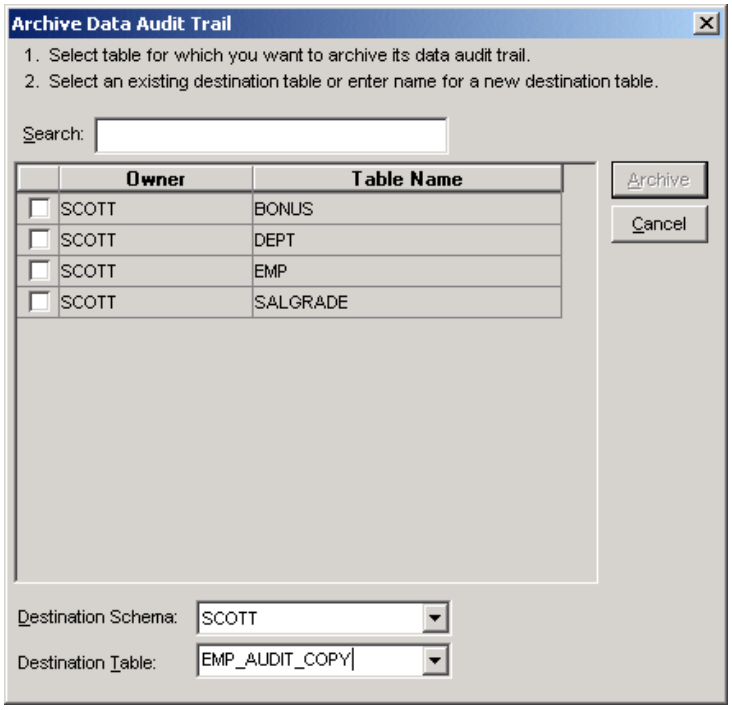
Note: Not all database systems support disabling table triggers. Please refer to Feature matrix by DBMS topic for additional information.

Data-change audit trail management

Archiving a data-change audit trail to a table

Select the **Data Audit/Archive to Table** command from the DB Audit Expert menu.

The **Archive Data Audit Trail** dialog appears. Select name of the existing audited table from the source table list.



Either select the name of an existing table or enter the name of a new table. DB Audit Expert automatically creates the destination table if it does not exist. Note that if have not purged the data audit trail since the last archiving operation, you will have archived the same audit trail records twice.

Repeat these steps to archive another data change audit table.

Archiving a data-change audit trail to a file

Select **Data Audit/Export to File** command from the DB Audit Expert menu.

The select export file dialog appears. Specify name of the file to which you want to archive the data audit trail. If you select an existing file, DB Audit Expert will overwrite this file.

The select table dialog appears. Select name of the table for which you want to archive the data audit trail.

If necessary, repeat described steps for another table being audited.

Truncating data-change audit trail

Select **Data Audit/Purge** command from the DB Audit Expert menu.

The Select Audited Table dialog appears. Select the name of the table from which you want to purge the data audit trail.

If necessary, repeat these steps for to purge data from another audit table.

Scheduling periodic audit trail purges

Due to the nature of auditing, volume of audit log data can grow quite quickly in a short period of time. DB Audit supports automatic data purge procedures that can purge old data from the audit trail tables.

Purging removes all records from the audit trail tables that are time-stamped before a certain date and time. Purging may be setup as a scheduled job that runs automatically at a specified time.

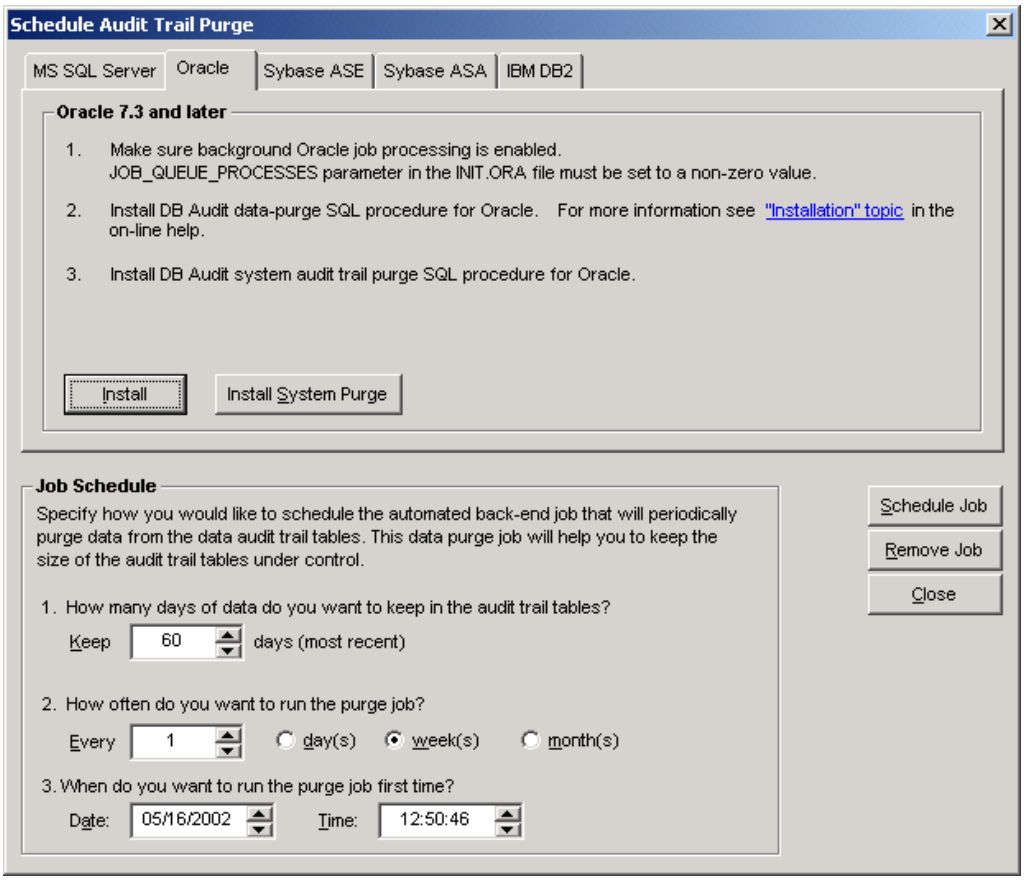
To install audit data purge procedure:

1. Click the **Tools/Schedule Periodic Purge** menu or click the **Schedule** button on the Toolbar.

2. Review the requirements for your database system as displayed on the screen. Make sure your database system has all the necessary components and settings for the data purge job.

3. Click the **Install** button to install the purge procedure.

    **Oracle:** Because Oracle does not support automated system audit trail purging (SYS.AUD$ table), DB Audit provides an additional procedure you can use to purge the system audit log. To install the system purge procedure, click the **Install System Purge** button.



DB2: In some environments, DB2 uses an external C compiler when compiling SQL stored procedures like the data audit purge procedure *DB_AUDIT.SP_AUDIT_PURGE*. To install this procedure successfully, make sure your DB2 server settings are properly configured so that DB2 can locate and use the right compilers. It might be necessary to set the following environment variables:

```
DB2PATH=C:\SQLLIB
DB2_SQLROUTINE_KEEP_FILES=yes
DB2_SQLROUTINE_COMPILER_PATH=your compiler bin directory
```

In addition, you may need to set the correct compiler in the *\SQLLIB\function\routine\sqlproc.mak* file. You should close all connections to the database before editing this file: sometimes even though you edit this file and save it, the changes are not saved as DB2 maintains a lock on the file if connections are active. If you perform all the operations above and still get error messages while installing the data audit purge procedure, re-open *sqlproc.mak* and make sure the changes you made were saved. Alternatively, try repeating the procedure after making sure that there are no open connections to the DB2 database. Also make sure this file's read-only attribute is not set.

To schedule the purge job:

Oracle, SQL Server: Because of the native database support for job scheduling available in Oracle and in Microsoft SQL Server, you can use the DB Audit GUI to set up the purge job.

3. Specify parameters for the purge job as instructed on the screen.

4. Click the **Schedule Job** button to create a new job or click the **Remove Job** button to remove an existing data purge job.

ASE, ASA, DB2: Most versions of ASE, ASA and DB2 do not support job scheduling using database server facilities. As an alternative, you can setup periodic runs of the DB Audit data audit purge procedure using either a standalone scheduling utility such as 24x7 Scheduler or an available host operating system scheduling utility..

For example, on Windows NT systems, you can use the *AT* command. On UNIX systems you can use *crontab* to schedule unattended run of a batch job using the *ISQL* or DB2CMD utility. Consult your database and/or operating system documentation for details on how to use the these available utilities to schedule batch or non-interactive mode job scheduling.

The name of the data audit purge procedure is *DB_AUDIT.SP_AUDIT_PURGE*.

Note: This procedure can be installed and run on systems that support dynamic SQL executed from within stored procedures (DB2 5.0 and later, ASE 12.0 and later).

Tip: To schedule *DB_AUDIT.SP_AUDIT_PURGE* stored procedure runs using the 24x7 Scheduler, you can either create a program job to run the *ISQL* utility or create database job that directly connects to the database and executes the appropriate SQL command.

Examples

3. To run the *DB_AUDIT.SP_AUDIT_PURGE* procedure in ASE using a **program job:**

   Create a new job and select the "program job" option. For the job command line, enter the following:

   ```
   isql [-S server] [-U user] [-P password] [-D repository] [-i inputfile]
   ```

   In this command, replace the *'server', 'user',* and *'password'* parameters with the Sybase ASE server name, user name and password you use to connect to the database. Replace the *'repository'* parameter with the name of the database you picked for the DB Audit repository tables. Replace the *'inputfile'* parameter with name of an existing text file. The input file must contain the following two lines of text:

   ```
   exec db_audit.sp_audit_purge @days = [n]
   go
   ```

   Replace [n] with the number of days of audit history you want to retain in the database. Select the desired job schedule and save changes.

4. To run the *DB_AUDIT.SP_AUDIT_PURGE* procedure in ASE using a direct **database job**:

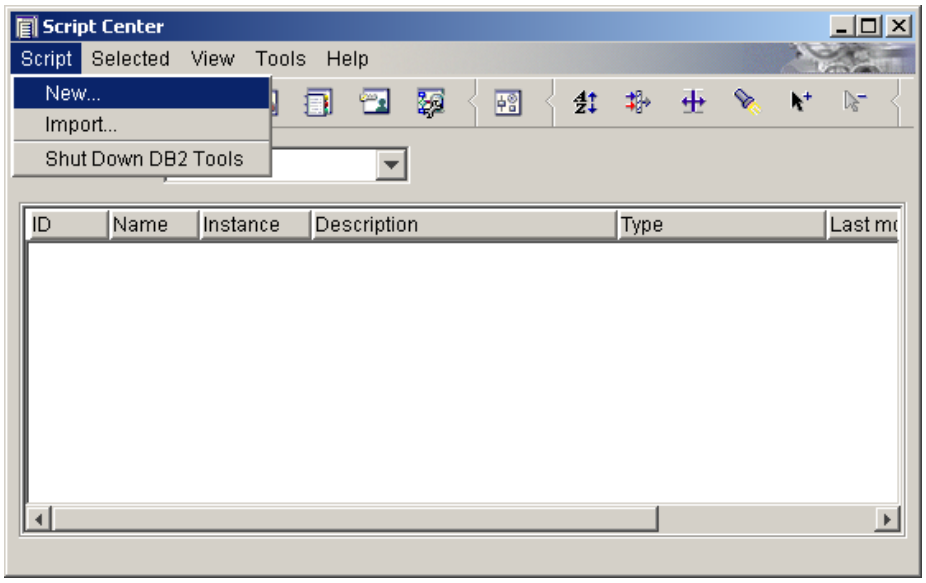
   Create a new database profile using the T**ools/Database Profiles** menu. Specify the required connection parameters and profile name (for example, "Sybase ASE"). Create a new job and select the "database job" option. For the job SQL enter the following:

   ```
   execute db_audit.sp_audit_purge @days = [n]
   ```

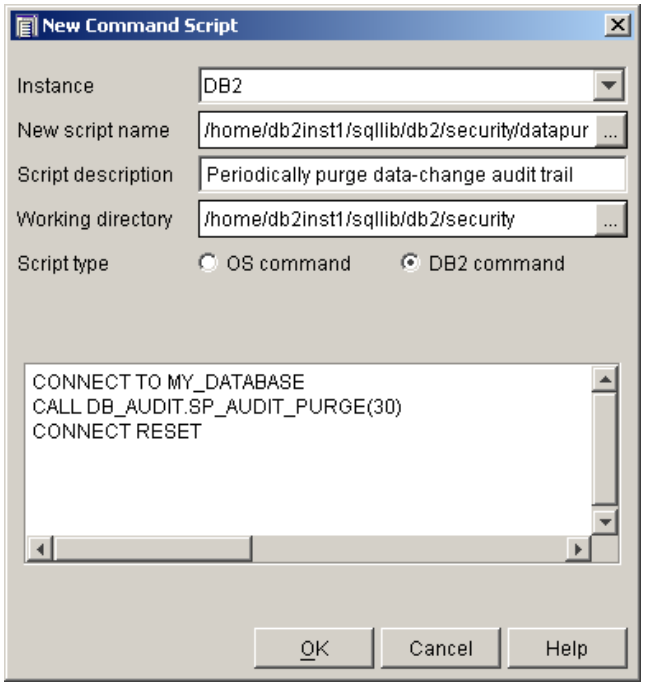
   Replace [n] with the number of days of audit history that you want to retain in the database. Choose the previously created database profile "Sybase ASE". Select the desired job schedule and save changes.

DB2: Some versions of DB2 support job scheduling using built-in database server facilities. In these versions, jobs can be scheduled using DB2 Script Center or DB2 Task Center tools. The following example demonstrates how to schedule data audit trail purge procedure using DB2 Script Center.

4. Start DB Script Center. Click **Script/New** menu to create a new job.



5. When the **New Command Script** dialog appears, fill-in the required job properties. You can choose any location for the script file and enter any description for the script. Example values are provided on following screenshot.



Make sure to select **DB2 command** for the **Script type** option.

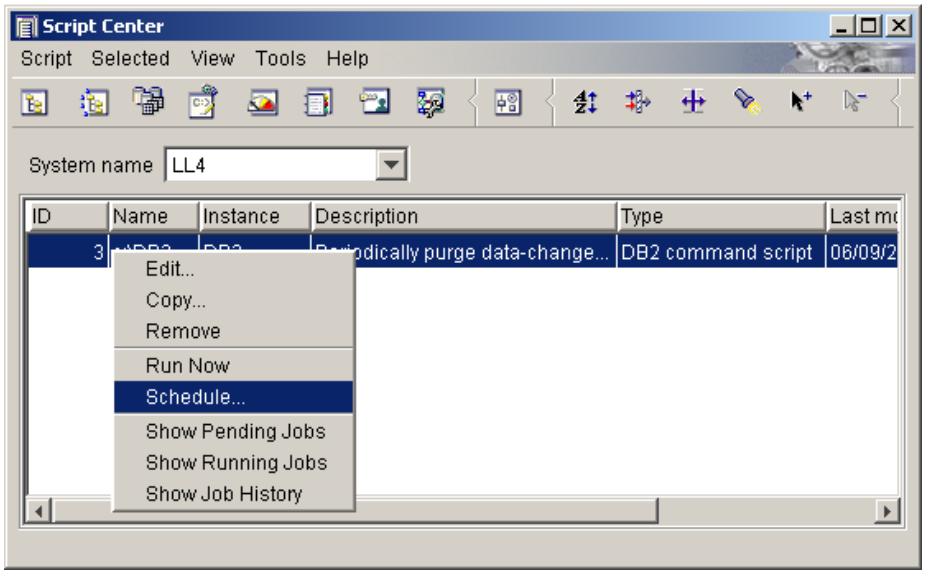
For the command script you must enter the following script

```
CONNECT TO MY_DATABASE
CALL DB_AUDIT.SP_AUDIT_PURGE(30)
CONNECT RESET
```

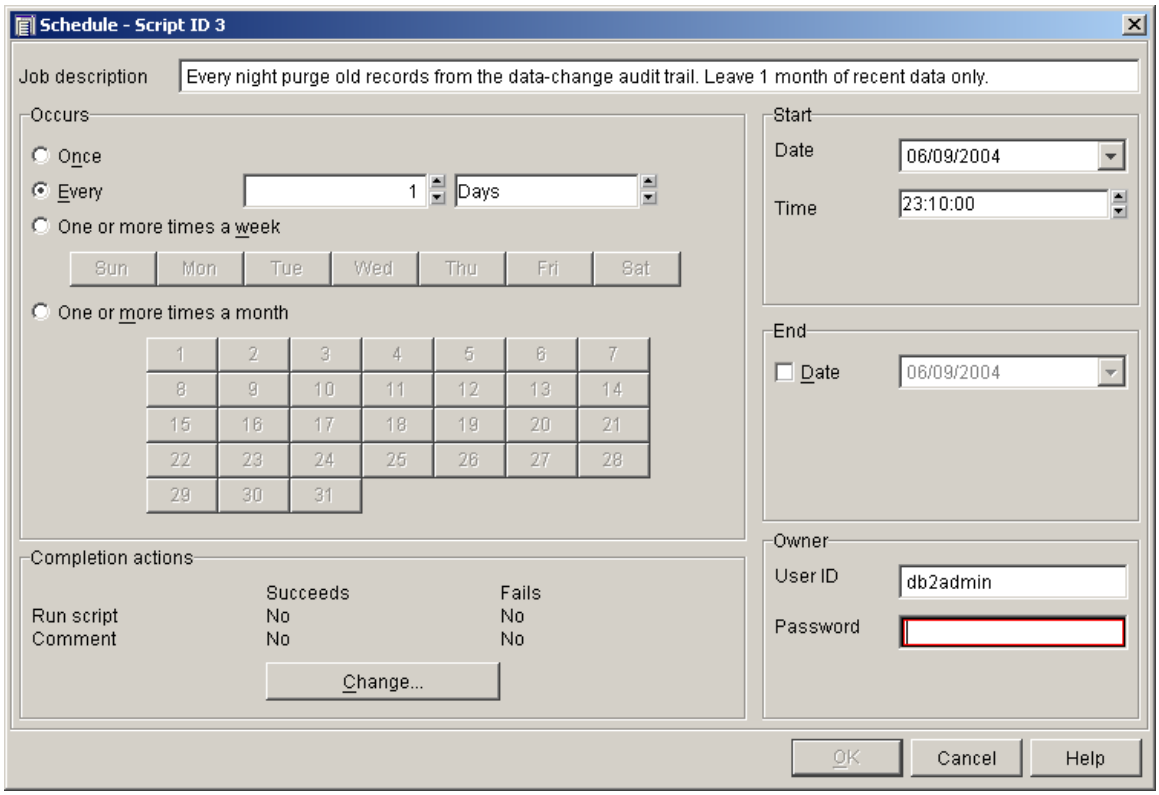
Note: In this script you must replace **MY_DATABASE** with the actual name of your DB2 database. Replace 30 with the number of days of audit history that you want to keep in the database.

Click the **OK** button to close the script dialog.

6. Now you need to schedule the created script. Right-click on the job line and then select **Schedule** from the popup menu.



This will open the **Schedule Script** dialog.



a) Enter a descriptive schedule name, for example, *"Every night purge old records from the data-change audit trail. Leave one month of recent data only."*

b) Enter the desired scheduling frequency. It is recommended to set this job to run once every date during "quiet" database hours.

c) Enter job initial start date and time.

d) Enter job owner information. This is the name and password of the user account that will be used to run the job. This account must have sufficient permissions to delete data from DB Audit repository tables.

e) Click the **OK** button to close the **Schedule** dialog.

That's it. The job is now scheduled and you can close the Script Center.

Scheduling periodic audit data trail archiving to files

DB Audit supports automatic archiving of audit trail data to operating system files. The same procedures are used to archive the system audit trail and data-change audit trails. For instructions on how to install and schedule these procedures, see Scheduling periodic audit trail archiving to files topic in CHAPTER 3.

Archiving audit trail to centralized audit repository

DB Audit supports automatic archiving of local audit trail data to a central repository system residing on a different database server.

See CHAPTER 8: Central Audit Repository for information on how to configure the central audit repository system and how to set up the Alert Center to periodically archive local audit-trail data to the central repository and then truncate the local audit-trail.

Before You Begin Data-Change Auditing

You should have CREATE USER, SELECT ANY TABLE, CREATE TABLE, CREATE TRIGGER, and REFERENCE ANY TABLE privileges.

 **DB2, Oracle:** You must have permissions to create tables in the default tablespace assigned to the DB_AUDIT user.

 **ASE, SQL Server:** You must have permissions to create tables in the default database assigned to the DB_AUDIT login.

 **ASE, SQL Server:** You must have permissions to create tables in the DB_AUDIT database and also ALTER permissions for the audited tables.

 **Note:** It is highly recommended that you log in as a database administrator to set up new data audit triggers. In SQL Server and ASE, use the SA account. In Oracle, use the SYSTEM or a similar account. In ASA, use DBA or a similar account, in DB2 and MySQL, use any appropriate administrator account.

CHAPTER 5: Vulnerability Assessment

Common Database Security Vulnerabilities

Overview

DB Audit provides several tools that can be used for assessment of common database security vulnerabilities. All these tools simulate generic database attack methods commonly used by hackers. The main purpose of these tools is to verify database and password security and to find users whose passwords are weak and can be easily guessed or cracked. The only tool specific to particular database types is the "Buffer Overflow Attack" tool. The purpose of that tool is to check for widely known programming flows in database servers caused by "buffer-overflow" vulnerabilities and to verify that your database server software has been patched correctly using recent service packs and security patches.

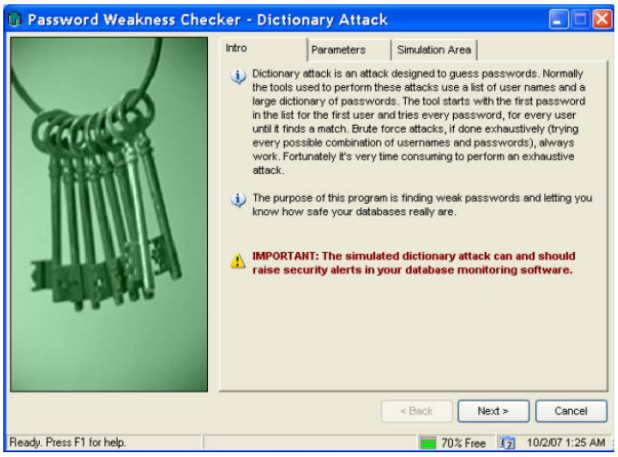
Use the **Tools/ Vulnerabilities and Penetration Testing Tools** menu to access database vulnerability assessment tools.

 **WARNING:** DB Audit license permits use of vulnerabilities assessment tools with databases running on your local network in development and quality assurance environments only! These tools may NOT be used for illegal or immoral purposes and use of them outside of your local network is strictly prohibited. Any abuse reported to SoftTree Technologies will result in the instant termination of your license. Use these database vulnerabilities assessment tools at your own risk; no warranties are given or implied. SoftTree Technologies will not be responsible for any damage caused by the use or misuse of these tools.

Password Weakness Checker– Dictionary Attack

Dictionary attack is an attack designed to guess passwords. Normally the tools used to perform these attacks use a list of user names and a large dictionary of passwords. The tool starts with the first password in the list for the first user and tries every password for every user until it finds a match.

Brute-force attacks are an alternative to dictionary attacks. If done exhaustively (trying every possible combination of usernames and passwords) and given enough time, brute-force attacks always work but, fortunately, they are very time consuming to perform. This is why hackers often choose dictionary attack as the most efficient method. The best defense against brute-force attacks is to enforce password length and complexity standards.



The "Dictionary Attack" tool uses dictionary attack method to simulate database dictionary attack.

Click **Tools/ Vulnerabilities and Penetration Testing Tools/Weak Passwords – Dictionary Attack** menu to start the tool.

The tool's wizard-like graphical interface is very straightforward and consists of only three pages:

1. The **Intro** page describes how the tool works. After you read the Intro page, click the **Next** button to advance to the second page.

2. The **Parameters** page provides a place to select the attack target and options.

   If you want to test a system whose profile you already have configured in the DB Audit database tree, you can use the **Profile** drop-down list to select that system. If the needed system is not listed, you can simply type a new name in the **Profile** box and then enter the connection driver and database server name in the **Driver** and **Server** fields.

   If you select ODBC as the connection driver, you must enter an ODBC data source name in the **Data Source** field. DB Audit will automatically obtain the target server name from the ODBC data source definition stored in the system registry.

   In the **Database Type** drop-down list, select the type of database you are going to test. Make sure to select the correct type. Based on the selection, the Dictionary Attack tool will use different user and password combinations commonly used with that database type.

   Check the **Try common user-password combinations** checkbox if you want the tool to try all known default and common user accounts and password for the selected database type. These default accounts can be found in many databases after installation of various database features and widely used software products, such as ERP applications, web servers and so on.

   Tip: DB Audit reads common user names and passwords from the tab-separated file **dictuser.txt.** This file contains over 700 pairs of user and password names. If you want to expand the user-password dictionary, you can add your own pairs to the end of the file.

   Check the **Try common passwords** checkbox if you want the tool to try all known commonly used or easily guessed passwords applied to all default database administrator accounts. Such easily guessed passwords include common dictionary words, blank passwords, user names with appended numbers, passwords same as user names, common keystrokes, and sequential letters and numbers.

   Tip: DB Audit reads common administrator passwords from the file **dictpass.txt.** This file contains over 2500 passwords. If you want to expand the password dictionary, you can add

your own passwords to the end of the file.

After you fill the attack properties, click the **Next** button to advance to the third page.

3. The **Simulation Area** page contains the attack workspace where DB Audit creates attacking sessions. It also displays progress of work and other status messages. The **Next** button on this page is replaced with the **Start** button. Click this button to begin the attack.

Depending on the selected attack options, performance of the computer running DB Audit and also performance of your database the attack time can take from several seconds to several minutes. Any correctly guessed passwords are displayed in the status area of the **Simulation Area** page. A message box providing complete list of all correctly identified passwords is also displayed at the end of the attack.

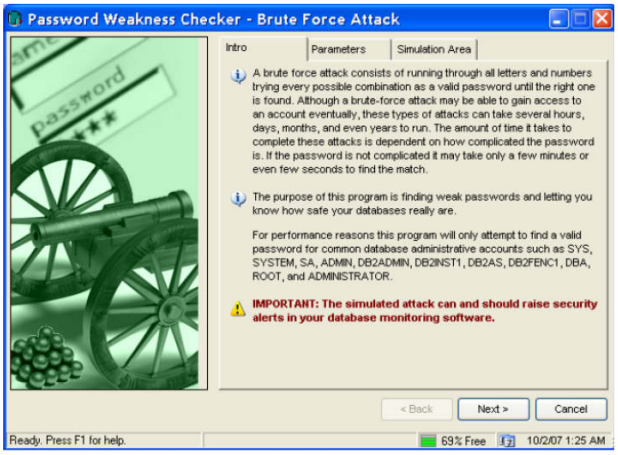
While the tool is running, it shows the current accounts, passwords, average password processing speed, elapsed and remaining time, and also total and processed number of passwords.

You can abort the attack at any time by clicking the **Cancel** button. It may take several seconds for DB Audit to abort processing.

Important Note: The simulated dictionary attack may trigger security alerts in your database monitoring systems. If security alerts are not raised during the attack simulation test, you should review your database and network monitoring systems and update them as necessary. Alternatively, you can use the DB Audit's Alert Center to setup proper database monitoring and alerting.

Password Weakness Checker– Brute-force Attack

A brute force attack consists of running through all letters and numbers trying every possible combination as a valid password until the right one is found. Although a brute-force attack may be able to gain access to an account given enough time, for passwords of sufficient length and complexity, these types of attacks can take several hours, days, months, or even years to run. The length of time it takes to complete these attacks is dependent on how complicated the password is. If the password is not complicated, it may take only a few minutes or even few seconds to guess it.



The Brute-force Attack tool uses brute-force method to simulate a database attack.

Click **Tools/ Vulnerabilities and Penetration Testing Tools/Weak Passwords – Brute-force Attack** menu to start the tool.

The tool's wizard-like graphical interface is very straightforward and consists of only three pages:

1. The **Intro** page describes how the tool works. After you have read the Intro page, click the **Next** button to advance to the second page.

2. The **Parameters** page provides a place to select the attack target and options.

   If you want to test a system whose profile is already configured in the DB Audit database tree, use the **Profile** drop-down list to select that system. If the needed system is not listed, you can simply type a new name in the **Profile** box and then enter the connection driver and database server name in the **Driver** and **Server** fields.

   If you select ODBC as the connection driver, you must enter an ODBC data source name in the **Data Source** field. DB Audit will automatically obtain the target server name from the ODBC data source definition stored in the system registry.

   In the **Database Type** drop-down list, select the type of the database you are going to test. Make sure to select the correct type. Based on this selection, the Brute-force Attack tool will try different administrative accounts used with that the selected database type.

   In the **Character Set** drop-down list, choose the character set that contains the alphabetic characters and numbers you want the tool to when generating passwords. You can choose from all Latin letters (note the case-sensitive option described below), all digits, all Latin letters and digits, or all Latin letters, digits and special symbols. Selecting larger character sets results in the tool generating more passwords which therefore requires more time to complete the attack.

    **Tip**: This option has direct impact on the password checking time. Richer character sets require the tool to try more passwords thus increasing the attack duration. The maximum number of passwords that can be generated using the chosen options is displayed on the bottom of the **Simulation Area** page.

   Check the **Case-sensitive** checkbox if your database uses case-sensitive passwords. If this option is not checked the Brute-force Attack tool generates all passwords in lower case.

    **Tip**: This option has direct impact on the password checking time. Case-sensitivity requires the tool to try a lot more passwords thus increasing the attack duration. The maximum number of passwords that can be generated using the chosen options is displayed on the bottom of the **Simulation Area** page.

   Specify a **password length** range using minimum and maximum values.

    **Tip**: This option has direct impact on the password checking time. Longer passwords require the tool to try more passwords thus increasing the attack duration. The maximum number of passwords that can be generated using the chosen options is displayed on the bottom of the **Simulation Area** page.

   After you enter the attack properties, click the **Next** button to advance to the third page.

3. The **Simulation Area** page contains the attack workspace where DB Audit creates attacking sessions and also displays names of standard administrative accounts for the selected database type, progress of work and other status messages. The **Next** button on this page is replaced with the **Start** button. By default, all administrative accounts are checked. If you want to narrow the scope of the attack and potentially decrease the time required to find a valid password, deselect the accounts you do not want to test or change the password generation options on the **Parameters** page.

   When ready, click the **Start** button to begin a new attack.

Depending on the selected attack options, the number of selected accounts, the performance of the computer running the DB Audit Management Console, and the performance of your database server,

the attack time may vary from several seconds to several years. If the administrator password is correctly guessed, the attack is automatically aborted for that account and a status message is displayed. DB Audit continues attacking the remaining accounts until all valid passwords are found or the selected character set is completely exhausted and all possible passwords have been tried.

Also, note that the Brute-force Attack tool begins the password guessing process using shorter passwords first. For example, if you set the minimum password length parameter to three and the maximum to six, the tool will start with three-character passwords, then try four-character passwords and so on.

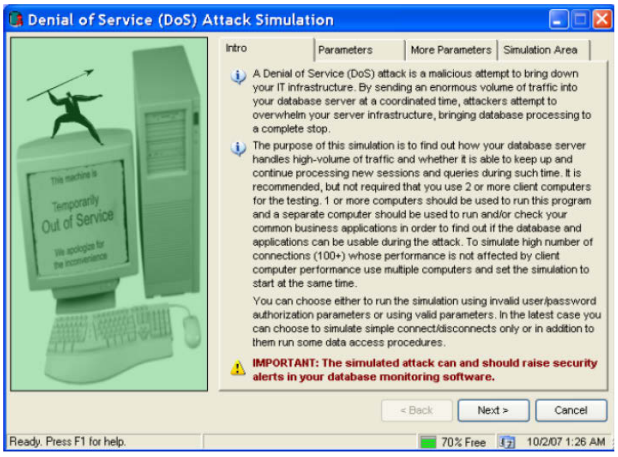
While the tool is running, it shows the current password length, as well as the current accounts, passwords, average password processing speed, elapsed and remaining time, and total and processed number of passwords.

You can abort the attack at any time by clicking the **Cancel** button. It may take several seconds for DB Audit to abort the processing.

> **Important Note:** The simulated brute-force attack may trigger security alerts in your database monitoring systems. If ecurity alerts are not raised during the attack simulation test, you should review your database and network monitoring systems and update them as necessary. Alternatively, you can use the DB Audit's Alert Center to setup proper database monitoring and alerting.

Denial of Service Attack

A Denial of Service (DoS) attack is a malicious attempt to bring down your database by flooding it with service requests. By sending an enormous volume of traffic into your database server at a coordinated time, attackers attempt to overwhelm your server infrastructure, bringing database processing to a complete stop.



The "DoS Attack" tool simulates multiple database applications attempting simultaneously to connect to your database and bring the database down or cause it to lock down which in turn may lead to the denial of service situation.

Click **Tools/ Vulnerabilities and Penetration Testing Tools/Generic Denial of Service Attack** menu to start the tool.

The tool's wizard-like graphical interface is very straightforward and consists of only four pages:

1. The **Intro** page describes how the tool works. After you read the Intro page click the **Next** button to advance to the second page.

2. The **Parameters** page provides a place to select the attack target and options.

   If you want to test a system whose profile is already configured in the DB Audit database tree, use the **Profile** drop-down list to select that system. If the needed system is not listed, type a new name in the **Profile** box and then enter the connection driver and database server name into the **Driver** and **Server** fields.

   If you select ODBC as the connection driver, you must enter an ODBC data source name in the **Data Source** field. DB Audit will automatically obtain the target server name from the ODBC data source definition stored in the system registry. If you want to simulate trusted connections check the **Trusted connections** checkbox.

   In the **Choose connection type** option group, select the type of the database connection you want to use for the attack. If you pick **Use valid database connections** option, you must enter valid user and password values in the **User** and **Password** fields. If you pick the **Use invalid database connections** option, DB Audit will use a randomly generated user name and password for each connection.

   Click the **Next** button to advance to the third page.

3. The **More Parameters** page provides a place to enter additional parameters for the DoS attack the attack.

   In the **Database Type** drop-down list, select the type of database you are going to test. Make sure to select the correct type. Based on this selection, the DoS Attack tool will try different connection optimizations appropriate for the selected database type.

   If you picked **Use valid database connections** option on the **Parameters** page, you can specify the behavior and desired activity of attacking sessions. You can choose either the **Connect and then immediately disconnect** option or the **Connect, perform-data access, and then immediately disconnect** options. The second option adds a data-processing function to the attacked sessions. This function can cause additional load on the database server, which can further affect your database's ability to accept new user sessions.

   Use **Attack Duration**, **Attackers** and **Start Time** parameters to control DoS attack properties.

    **Tip**: DoS attack efficiency is greatly affected by performance of the computer running the attack and by the bandwidth of your network. Do not attempt to simulate running hundreds or thousands of attacking sessions from a single computer. Instead use multiple computers to run the attack with a smaller number of sessions set up on each computer all using the same or incremental start time. This way you can simulate a more realistic real-world DoS attack and check your defenses.

   After you enter the required attack parameters, click the **Next** button to advance to the last page.

4. The **Simulation Area** page contains the attack workspace where DB Audit creates attacking sessions and displays progress of work messages. The **Next** button on this page is replaced with the **Start** button.

   When ready, click the **Start** button to begin a new DoS attack.

    **Tip**: If the start time selected on the **More Parameters** page is less or equal to the current system time, DB Audit starts attacking the database server immediately. Otherwise it

prepares for a new attack sessions and waits for the specified start time.

While the tool is running, it displays the current attacking sessions, their processing states, elapsed and remaining time, and the total number of processed connection requests. All these statistics are recorded in a work file. At the end of the run, a prompt is displayed asking whether you want to see charts demonstrating how the database server reacted to this attack. If you choose “YES,” the following charts will be displayed:

1. **umber of successful and failed connections as a function of time** – This chart shows how many connection attempts the attacking computer has been able to make and the database server has been able to process during the run. In an ideal situation, if the database server has not been affected at all, the chart should display a straight line rising from zero to the total number of connection attempts. If you have chosen to run the attack using the "invalid database connections" option, you should see just one line indicating the number of failed connections only. If you have chosen to run the attack using the "valid database connections" option, you should see two stacked lines indicating both successful and failed connections.

2. **Average connect time as a function of time** – This chart shows how long it took the database server to process connection attempts made from the attacking computer. The connect time is expressed in milliseconds. If you have chosen to run the attack using the "invalid database connections" option, the chart displays the average time it took the server to deny a connection. If you have chosen to run the attack using the "valid database connections" option, the chart takes into account two different times:
   - The average time for failed connections (time the database server needed to deny a connection or the client to abort a timed out connection attempt)The average time for establishing a successful connection

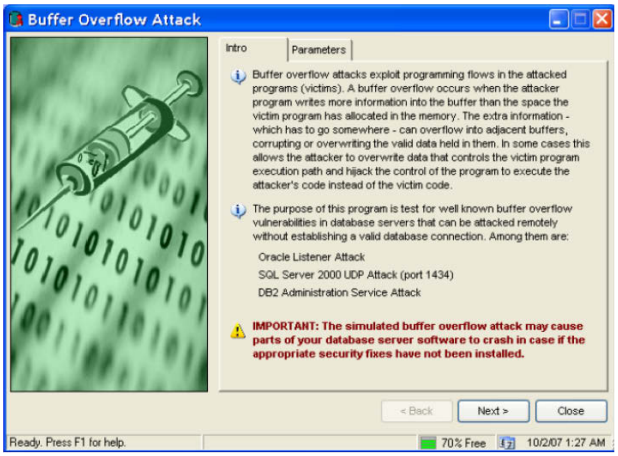
3. **Average query time as a function of time** – This chart shows how long it took the database server to process data access queries sent from the attacking computer. The query processing time is expressed in milliseconds. This chart is only available if you chose to run the attack using the "valid database connections" option and also selected the "connect, data-access, disconnect" option.

You can abort the attack At any time by clicking the **Cancel** button. It may take several seconds for DB Audit to abort the processing.

Important Note: The simulated DoS attack may trigger security alerts in your database and/or network monitoring systems. If security alerts are not raised during the attack simulation test, you should review your database and network monitoring systems and update them as necessary. Alternatively, you can use the DB Audit's Alert Center features to setup proper database monitoring and alerting.

Buffer Overflow Attack

Buffer overflow attacks exploit programming flows in the attacked programs (victims). A buffer overflow occurs when the attacker program writes more information into a buffer than the space the victim program has allocated in memory. The extra information may overflow into adjacent buffers, corrupting or overwriting the valid data stored in them. In some cases, this allows the attacker to overwrite data that controls the victim program's execution path, taking control of the program and executing the attacker's code instead of the victim's code.



Important Notes:

- The purpose of the Buffer-Overflow Attack tool is to verify that your database server software has been patched correctly using recent service packs and security patches. Do not assume that If the tool is unable to exploit these vulnerabilities, your databases are safe. You should check with your database vendor for recent security bulletins and install recent database security fixes as they become available.
- The simulated buffer overflow attack may cause parts of your database server software to crash if the appropriate security fixes have not been installed.
- The Buffer Overflow Attack tool sends specially crafted data packets to the database server, attempting to cause buffer overflows and disruption of the normal database processing. These packets do NOT contain any viruses or other virus-like payloads and do not directly affect any systems other then the attacked database server.

Here is a list of links to database vendor sites where you can find the most recent security bulletins, advisories and updates:

SQL Server: http://www.microsoft.com/technet/security/prodtech/SQLServer.mspx
Oracle: http://www.oracle.com/technology/deploy/security/alerts.htm
DB2: http://www-306.ibm.com/software/sw-bycategory/subcategory/SWB30.html
ASA, ASE: http://downloads.sybase.com/swd/base.do?client=support
MySQL: http://dev.mysql.com/downloads/

DB Audit currently supports buffer overflow attack simulations for the following widely known vulnerabilities:

- Oracle listener buffer overflow (un-patched versions 8i, 9i, 10g, all platforms) - TCP port 1521
- SQL Server 2000 buffer overflow in name resolution service (SP1 and SP2) – UDP port 1434
- DB2 UDB for Linux, Unix and Windows buffer overflow in DAS service (un-patched versions 7 and 8) - TCP port 523

Click the **Tools/Vulnerabilities and Penetration Testing Tools/Buffer Overflow Attack** menu to start the tool.

The tool's wizard-like graphical interface is very straightforward and consists of only two pages:

4. The **Intro** page describes how the tool works. After you read the Intro page, click the **Next** button to advance to the second page.

5. The **Parameters** page provides a place to select the attack target and options.

   If you want to test a system whose profile is already configured in the DB Audit database tree, use the **Profile** drop-down list to select that system. If the needed system is not listed, type a new name in the **Profile** box and then enter the connection driver and database server name in the **Driver** and **Server** fields.

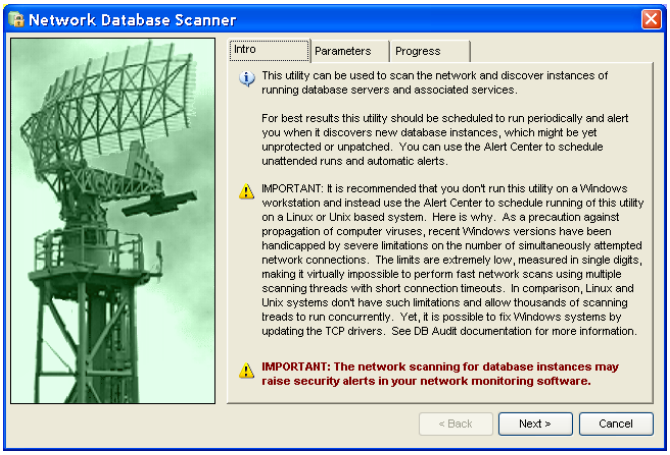
   > **Note:** Regardless of what you have selected in the **Driver** drop-down list, you must enter the IP address or name of the server you want to test in the **Server** field.

   In the **Database Type** drop-down list, select the type of database you are going to test. Make sure to select the correct type. Based on this selection, the Buffer Overflow Attack tool will try different attack parameters and methods appropriate for the selected database type.

   The **Next** button on this page is replaced with the **Start** button. Click the **Start** button to begin a new buffer overflow attack.

Network Database Scanner

The Network Database Scanner tool can be used to discover hidden instances of database servers on networked computers. The tool scans the network in a specified IP address range or IP addresses list, looking for database servers that accept network connections on well-known or custom ports. The scanner tool analyzes received responses and, based on the contents of received data packets, it determnes the type, version, authentication mechanism and other parameters associated with each found database server instance. The scanner tool is also capable of finding various server management and broadcasting services, such as DB2 Administration Server or SQL Server Browsing Services . The tool can be run in interactive graphical It can also be scheduled for automatic daily or weekly runs using the Alert Center facility.



Note about running the Network Database Scanner on Windows-based systems: As a precaution against rapid distribution of computer viruses, starting with Windows XP SP2, Microsoft put a severe limitation on the speed and number of simultaneous TCP connections. These limitations prevent efficient working of security scanners like the Network Database Scanner tool which, in turn,

severely degrades the scanning speed. In many cases, connections may timeout before a response is received from the database server. As a result, scanning is slow and is not completely reliable.

In comparison, running the Network Database Scanner tool on a Linux, Unix or Mac computer is much more efficient. In fact, the scanning speed is at least **10000 times faster** on these types of systems.

However, it is possible to unlock handicapped TCP/IP functions on Windows-based systems and restore their original networking capabilities. This is accomplished by patching the network driver for TCP/IP protocol, the TCPIP.SYS file. If you plan to run the Network Database Scanner tool periodically on a Windows-based host, you should install one of the available patch solutions available from a number of third party vendors. For example, you can install the well-known TCP/IP driver patch known as **EvID4226Patch.exe.** This patch is not shipped with DB Audit. Search the Internet for a download link.

Running Network Database Scanner In Interactive Mode

The following describes how to use the Network Database Scanner in interactive mode.

1. Click **Tools/ Vulnerabilities and Penetration Testing Tools/Network Database Scanner** menu to start the tool in graphical mode. This will display the tool's wizard-like graphical interface, which is very straightforward and consists of only three pages. The first **Intro** page describes how the tool works. After you read the Intro page click the **Next** button to advance to the second page.

2. The **Parameters** page provides a place to select the scanning targets and options. Select the types of database targets you want to discover on your network.

3. For each selected target type, specify the list of **port numbers** or **port ranges** you want the scanner to search for on the selected targets. Port numbers and ranges must be entered as comma-separated lists; spaces after commas are optional. Insert a dash between port number values to specify a range of values. You can mix ranges and individual port numbers for each type of target. The following formalized specification describes the format for port number specifications. Several examples are also given.

   Port_Spec = *Port* | *Port_Interval*
   Port_Interval = *Port-Port*
   Port=an integer from 1 to 65535 inclusive

   Examples:

Port spec	Meaning
50000	Port # 50000
1443-1532	Ports from 1443 to 1532 inclusive
50000,1443-1532	All Ports listed above (**comma**-separated)

4. Specify the list of IP addresses and/or address ranges you want to scan for the selected targets. The list can contain both individual addresses and ranges; spaces after commas are optional. To specify a range of addresses, insert a dash symbol between the first and last address in the range. The following formalized specification describes the format for IP addresses numbers. Several examples are also given.

   IP address spec is a way of specifying a number of legacy (IPv4) IP addresses or ranges of IP addresses in a compact text form.

IP_Address_Spec = IP_Address_Spec [*IP_Address_Interval*]
IP_Address_Spec=Octet_Spec.Octet_Spec.Octet_Spec.Octet_Spec
Octet_Spec=Octet | *Octet_Interval*
Octet_Interval=Octet-Octet

Octet=an integer from 0 to 255 inclusive

Examples:

IP spec	Meaning
192.168.33.11	IP address 192.168.33.11
192.168.34.43-132	IP addresses from 192.168.34.43 to 192.168.34.232 inclusive
9-12.168.35.78	IP addresses 9.168.35.78, 10.168.35.78, 11.168.35.78 and 12.168.35.78
192.168.41-57.0-255	IP addresses from 192.168.41.0 to 192.168.57.255 inclusive
192.168.33.11, 192.168.34.43-132, 9-12.168.35.78, 192.168.41-57.0-255	All IP addresses listed above (specified as a comma-separated list or address ranges and individual addresses)

5. Choose additional scanning properties.

   Tip: The default properties are pre-configured for use with Windows systems running non-patched TCP/IP drivers. On non-patched Windows systems, the value for "concurrent scanners times concurrent connections" should not exceed 10. On patched Windows systems and also on Linux , Unix and Max systems, this value can be set to a very high number, for example 10000.

   The **Concurrent Scanners** property controls the number of concurrent network scanner threads used by the tool to search for the database servers and services.

   The **Connection Timeout** property sets the maximum period amount of time the database server will wait for a response before ending the connection. On one hand, smaller values could increase overall scanning speed, but smaller values may decrease the accuracy of scans and cause the scanner to miss some servers.

   The **Maximum Concurrent Connections** property controls the maximum number of concurrent non-blocking network connections that each scanning thread can open simultaneously. Higher numbers increase overall scanning performance; however, if the system has a set limit on the number of concurrently open TCP/OP connections, higher values can cause the scan to fail and connections to be blocked on the scanner side before the scanner gets a chance to receive a response from a database server.

6. Click the **Next** button to advance to the third page and start the scan.

   While the tool is running, it displays current scanning progress for each selected target and for overall scan progress. You can abort scanning at any time by clicking the **Cancel** button. It may take several seconds for DB Audit to abort the processing.

Displaying, Saving, and Printing Scan Results

Scan results are refreshed automatically in real time as the tool discovers new database servers on the network. Results begin appearing in the **Found Database Servers** window as soon as the first

server is located.

When the process completes, you can save or print the results.

To save the report in text, XML, Excel, or another supported file format, click the **Save** button located above the scan results. The **Save** dialog appears. In the **File Type** drop-down menu, select the required file type and click **OK**.

To print the displayed results as a report, click the **Print** button above the scan results.

Running the Network Database Scanner in Non-Interactive Mode

You can schedule non-interactive network scans. The non-graphical version of this tool is available in DB Audit API and available with Alert Center component.

The version shipped with the Alert Center automatically scans all IP addresses on the home network using the default port numbers for all types of database servers. If the scan finds a database server instance that has been added since the last run, a the scanner automatically generates a report notifying you of the discovery.

Use the following method to schedule non-interactive Network Scanner runs using the Alert Center:

1. In the DB Audit Management Console main menu, click **Tools/Alter Center** command. This will start the Alert Center Remote Console.
2. Connect to the Alert Center.
3. Click the **Center Audit Repository based Alerts and Reports** top-level item displayed in the Alerts and Reports browser. Follow the prompts displayed by the Wizard.
4. Click the **Alerts / New Report** command in the top-level menu. The Report Configuration screen appears.
5. There is nothing to choose on the first configuration screen. Click the **Next** button to advance to the next step.
6. In the **Reports** drop-down list, select **Discover New and Hidden Database Servers (Network-wide scan)** item, then click the **Next** button again.
7. Choose the desired report schedule and recipient, then click the **Finish** button.

Notes: In non-interactive mode, you cannot pick individual scan options for individual servers. The network scanner tool automatically scans for all types of servers using their default port numbers.

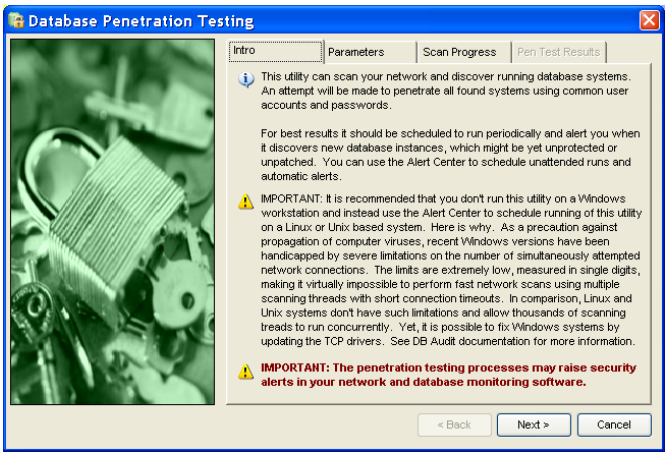
If you need to use custom settings for different servers, consider licensing the DB Audit API. The API allows running scans using single-line command line batch files with command line parameters. You can also use the API to write sophisticated programs that take full advantage of the programmatic API functions.

Database Penetration Testing

DB Audit's Database Penetration Testing tool mimics the actions of real life attackers. Conducting

periodic penetration tests is a valuable practice to use in evaluating your system security and preparing your defenses against real life attackers.

The Database Penetration Testing tool performs a two-step test: first, it scans your network for all instances of database servers running on networked computers, then it uses a data-dictionary attack to attempt to find common database accounts and passwords. Essentially, the Database Penetration Testing tool is a combination of the Network Database Scanner tool and the Password Weakness Checker– Dictionary Attack tool. The Database Penetration Testing tool automates the entire process of searching for database servers on the network and performing penetration testing on those servers that it discovers. The results of the penetration are displayed on the penetration report



For more information on supported network scanning options and limitations, read the Network Database Scanner topic.

For more information on supported data-dictionary methods and customizing the data dictionary, read the Password Weakness Checker– Dictionary Attack topic.

CHAPTER 6: Alerts

The Alert Center

The DB Audit enterprise license includes the advanced Alert Center, which automates the difficult and time consuming task of checking and analyzing database audit trail records. It can be also used to automate the process of archiving audited databases to a central audit repository server. The following paragraphs describe Alert Center capabilities:

- The Alert Center analyses audit trail data for patterns of activity that are either clear security violations or just suspicious, intrusive or anomalous (in other words, do not correspond to normal user activity) and alerts the system administrator to such activity.
- The Alert Center allows administrators to define automated countermeasures called "incident response jobs," which can be used for suspending or terminating processes, locking or terminating user sessions, shutting down and restarting database servers, and executing operation system scripts and commands.
- The Alert Center runs scheduled reports and automatically emails generated reports to specified recipients. DB Audit provides a number of prebuilt reports and alerts that come with the Alert Center. You can also create custom reports and alert monitors to fit your specific business requirements and run them using the Alert Center.
- The Alert Center automates audit data archiving processes for moving audit trail data from audited databases to a central audit repository server.
- The Alert Center automates audit space management for a central audit repository server.

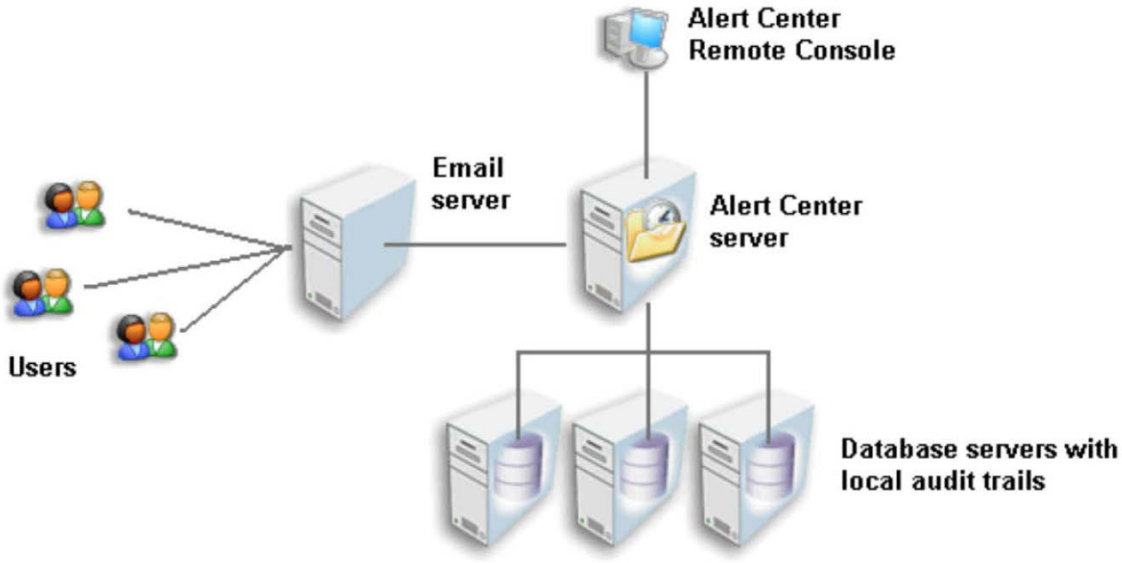
How It Works

The Alert Center uses SQL queries to query and analyze data in the audit trail tables. It searches for specific events, activities or activity patterns that have exceeded their threshold and are therefore in a state of alarm or warning. You set up thresholds using the Alert Center Remote Console, which can be invoked from the DB Audit Management Console or directly from the Windows **Start** menu.

In the event a problem is detected, the Alert Center generates an email alert and sends it to designated administrators. It can be also configured to automatically run a "fix it" job, called an accident response job. Response jobs can be used to start batch files, run shell scripts, execute database commands and so on.

The Alert Center provides the flexibility to schedule the evaluation of audit trail data at specified intervals using specified queries and parameters. It is a useful feature since you can schedule resource intensive audit trail monitoring jobs during long intervals while scheduling light jobs using short intervals.

The following diagram explains how the Alert Center works when a central audit repository is not used.



Alert Center Remote Console is a tool you use to remotely manage audit trail monitoring jobs that are scheduled and run using the Alert Center. These jobs are called alerts. Using this tool, you can create new alerts or modify, delete, enable or disable existing alerts. Multiple users can connect to and manage the Alert Center concurrently.

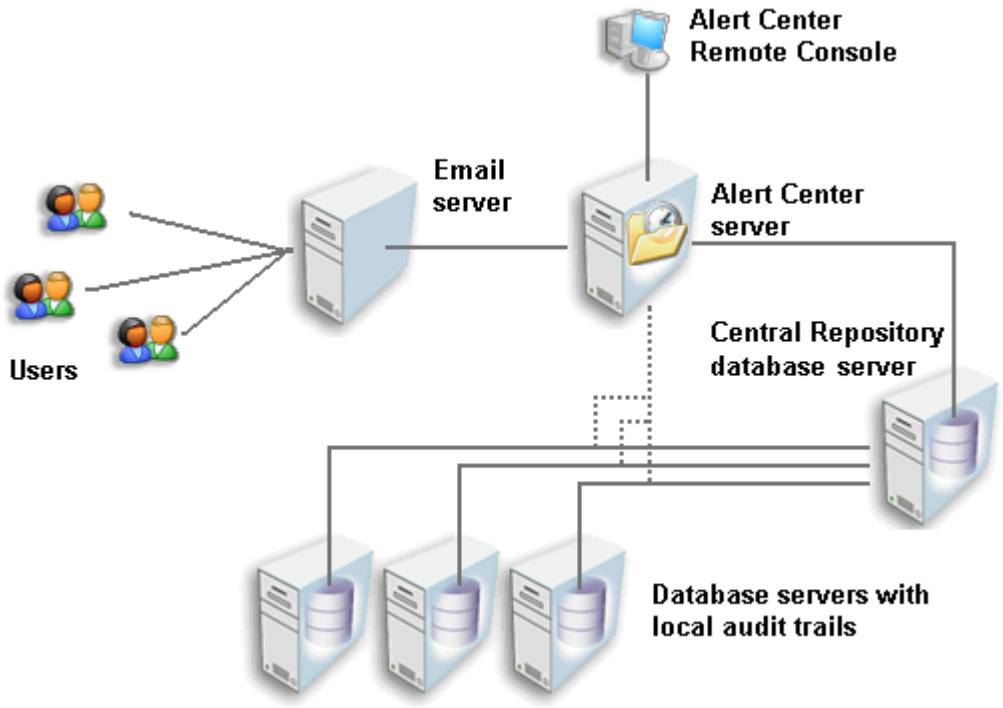
Alert Center server is a watchdog system responsible for scheduling and running audit trail monitoring jobs and generating alerts. The Alert Center can be set up to run on a standalone server or workstation, or it can be set up to run on any other computer involved in the data processing such as your database server. The Alert Server must run 24 hours a day 7 days a week in order not to miss important alerts.

Email Server is the existing email server you use to send and receive email messages.

Users are the alert recipients specified in alert properties. You can specify different recipients for different alert types and databases.

Database servers are the database servers on which you have installed and enabled auditing.

The following diagram explains how the Alert Center works when central audit repository is used.



Central Repository database server is the database server hosting the central audit repository database. The Alert Center server runs audit trail archiving jobs that pull data from local audit trails of the audited database systems to the central repository database.

All other components are the same as on the previous diagram.

Alert Center Server

The Alert Center is a multi-purpose tool. The main purpose of the Alert Center server is to periodically run audit trail monitoring jobs, generate alerts, and optionally invoke automated countermeasure handlers when security violations or other intrusive or suspicious activities are detected in the database.

The following prerequisites must be met to use the Alert Center server:

- For using the Alert Center, the 24x7 Scheduler Alert Center Edition must be installed and run on a computer in your network. See Alert Center Server Installation topic in CHAPTER 15 for more information on how to install and configure the Alert Center server.
- One or more database systems must be configured for system auditing. For information on how to enable the system-level audit trail, read CHAPTER 3, System Auditing.
- Proper JDBC or ODBC drivers and, if necessary, database client software must be installed on the computer running the Alert Center so that audit trial monitoring jobs can successfully connect to your databases. Note that Oracle, IBM and Sybase provide ODBC and JDBC drivers on their database server and client software installation CDs. Microsoft provides an

ODBC driver for Windows on the installation CD. The JDBC driver for SQL Server can be freely download and installed from the Microsoft web site. This driver is designed to work on both Windows and Unix systems.

See your database documentation for information on how to install the required JDBC or ODBC drivers.

- A network connection must be available between the Alert Center computer and the monitored databases.
- A network connection must be also available between the Alert Center computer and the computer running DB Audit Management Console.

The following paragraphs describe the process flow from alert configuration to alert generation and delivery:

1. Using the Alert Center Remote Console, you define database connections between the Alert Center server and the monitored databases for which you have previously configured and enabled system-level audit trail. For information on how to enable system-level audit trail, read CHAPTER 3, System Auditing.

    **Tip**: The Alert Center Remote Console is part of DB Audit Management Console software that can be installed on any Windows-compatible computer. Although the Alert Center supports a web-based graphical management interface, it is highly recommended that you use the Alert Center Remote Console, which is specifically designed and optimized for creating and managing alerts. The Alert Center Remote Console can be used to remotely manage DB Audit Alert Center.

2. Using the Alert Center Remote Console, you define new alerts and modify existing alerts and their parameters such as thresholds, scopes, frequencies, and so on. For each alert, you must also define alert recipients. For example, you could define a *"Connection Attempts from Terminals Not in Your Network Domain"* alert that runs every 30 minutes, and that analyzes the last 60 minutes of data recorded in the *Production* database audit trail using a threshold value of 10 attempts. You also assign a *name@company.com* email address as an alert recipient.

3. Once you have at least one alert defined, the Alert Center server starts running the alerts as specified in the alert parameters. Using the example from the previous paragraph, every 30 minutes the Alert Center connects to the *Production* database and executes an SQL query specific to the *"Connection Attempts from Terminals Not in Your Network Domain"* alert type. This query analyzes data from the system audit trail looking at records having timestamp values from 60 minutes prior to the current time and that have a value in the domain column that does not match your network domain. After SQL query execution, the Alert Center disconnects from the database.

4. If the audit trail monitoring job finds a problem in the audit trail data, the Alert Center generates an alert, logs it to the job log, and then immediately emails it to the designated alert recipient. Using the example from paragraph 2 above, the Alert Center checks to see if there are more then ten connection attempts and, if there are, it generates and emails the alert.

   If an "incident response jobs" job has been assigned to the alert, the Alert Center executes that job.

5. If the audit trail monitoring job fails for any reason such as that a database connection cannot be established, the Alert Center sends an email notification to the alert recipient with a description of the problem.

Audit Trail Monitoring Jobs and Email Alerts

When a Central Audit Repository Server Is Not Used

The Alert Center maintains a separate job queue for every configured database system and connection. All monitoring jobs for the same database are run in the same queue created for that particular database. Each job queue is independent of other queues, and because of that, monitoring of jobs for different database systems is also completely independent with the result that all jobs can run concurrently. Multiple monitoring jobs for the same database system are run sequentially based on their scheduling times and priorities.

When a Central Audit Repository Server Is Used

The Alert Center uses the dedicated [default] job queue to run audit trail monitoring jobs. All audit trail archiving jobs run in their own queues with a separate queue created for each monitored database system. Multiple monitoring jobs for the central repository system run sequentially based on their scheduling times and priorities.

Each monitoring job automates checking of database audit trail records for a single type of security-related event. Separate jobs must be scheduled for different event types and for separate audit trails (e.g. database systems).

The Alert Center gives you the flexibility to assign the same or different alert recipients to different alerts. For example, you can assign different security officers and database administrators to alerts generated for different databases.

If you want to send an alert to multiple recipients, specify an email distribution group common to all alert recipients whom you want to get the alert.

Different alert types generate different alert messages. Alerts dealing with generic problems; in other words, those that use threshold parameters, simply notify alert recipients about detected security breaches or suspicious activities. Information included in these alerts includes time of alert generation, alert name and type, and the database system where alert conditions have been found. Other alerts, such as alerts created to monitor access to specific objects after regular business hours, send summary reports and detailed reports sent as email attachments. These alerts include detailed information on the nature of the suspicious activity, the time it occurred, and the user associated with the security or system activity event.

DB Audit supports a variety of predefined alert types and allows you to create your own custom alerts. Read the Supported Alert Types topic for more information about predefined alert types and how to create custom alerts.

Incident Response Actions

An effective intrusion detection and response policy defines actions to be performed by the automated countermeasure handler for incident response.

By default, the Alert Center generates an email alert for each detected incident. In addition, each audit trail monitoring job can be linked to one or more user-defined incident response jobs, which in turn, can be used to suspend or terminate suspicious processes, lock or terminate user sessions, shut down and restart database servers and so on.

Incident response jobs can be created as external batch jobs, as SQL queries, or as JavaScript

programs automatically invoked when triggered by an incident. Response jobs can be invoked either locally on the computer running the Alert Center or remotely on the computer running your database servers.

For more information and an example of a response job, see Creating Incident Response Jobs topic later in this chapter.

Tip: Do not confuse remotely invoked jobs running on database server computers with SQL jobs invoked locally but processed by the database server software running on another computer. The first type refers to jobs whose operating system processes are started on the computer other then the computer running the Alert Center. The second type refers to jobs that are started on the Alert Center computer, that connect to a database server running on another computer and that send one or more SQL commands to the database server for execution on the database server.

24x7 Scheduler Alert Center Edition software is the main component of the Alert Center. The 24x7 Scheduler provides a way to execute jobs on remote computers. If you do not have the Alert Center installed on your database server computer, you can still execute response jobs remotely on the database server computer provided you have a remote processing agent installed there. Remote processing agents allow response job created on the Alert Server computer to be executed by the agents on the database server computer.

For complete information on all available functions and features, see 24x7 Scheduler User's Guide.

Alert Center Remote Console

Overview

The Alert Center Remote Console is a tool you use to remotely manage audit trail monitoring jobs that are scheduled and run using the Alert Center. These jobs are called alerts. Using this tool you can create new alerts and modify, delete, enable or disable existing alerts.

The following prerequisites must be met in order to use the Alert Center Remote Console:

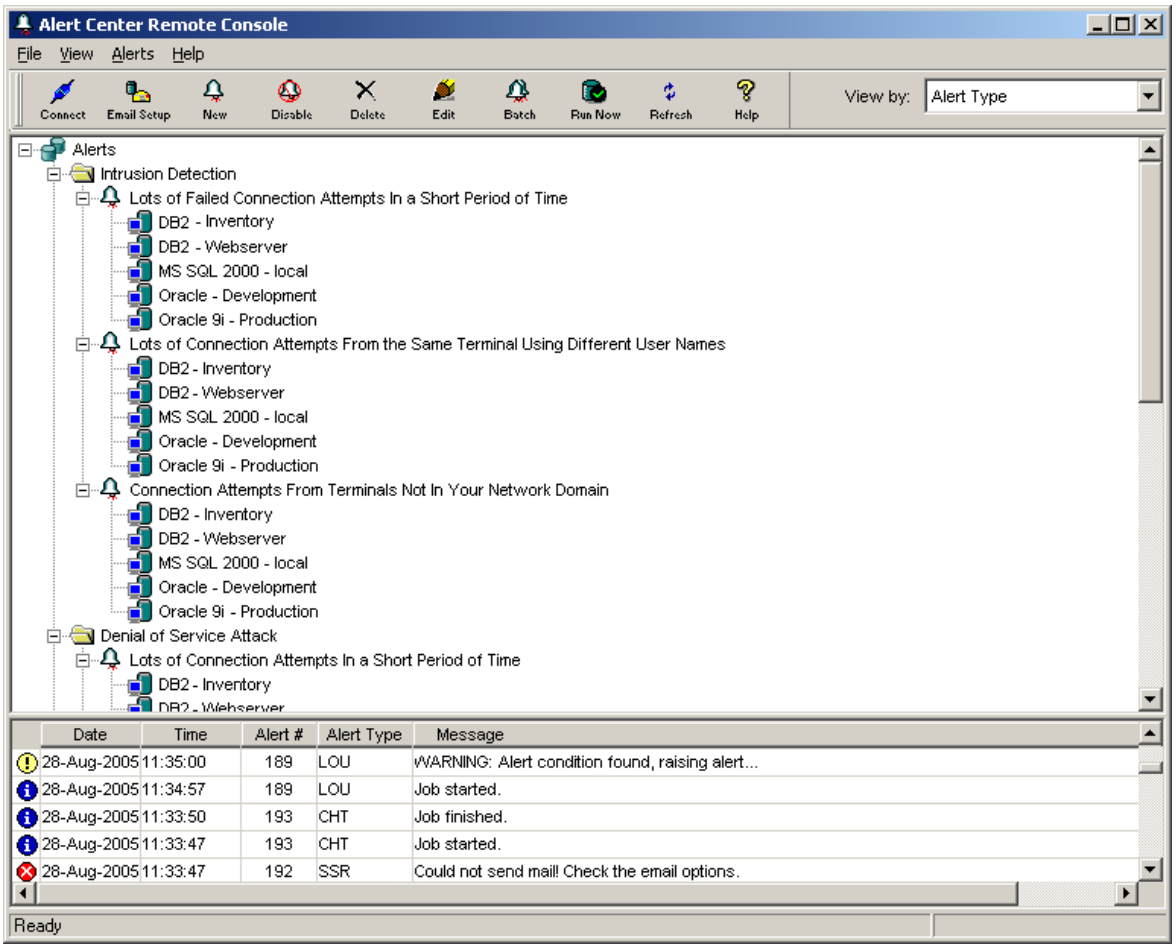
- The DB Audit Management Console must be installed on your workstation.
- The Java Run-time Environment (JRE) or Java Development Kit (JDK) version 1.4 or better must be installed on your workstation. If you don't have the correct version of JRE or JDK installed on the system, visit Sun Microsystems web site http://java.sun.com where you can you can freely download JRE and JDK software.
- The JAVA_HOME system environment variable must be defined on your computer. This variable must point to the directory where your JRE or JDK is installed; for example, *C:\Program Files\Java\j2re1.4.2_05*.
- For using the Alert Center Remote Console, the Alert Center server must be installed and run on a computer in your network. See Alert Center Server Installation topic in CHAPTER 15 for more information on how to install and configure the Alert Center server.
- A network connection must be also available between the Alert Center computer and the computer running DB Audit Management Console.

The Alert Center is a true client-server system. Multiple people can connect to and manage the Alert Center concurrently using The Alert Center Remote Console running on their workstations.

To launch the Alert Center Remote Console, either use Tools/Alert Center menu in the DB Audit Management Console or use the Alert Center shortcut on the Windows Programs start menu, DB

Audit's program group.

Below is an example screenshot of the Alert Center Remote Console.



The Alert Center Remote Console screen is divided into two panes:

- The top pane displays configured alerts and database systems.
- The bottom pane displays messages from Alert Center's alert processing logs. These messages provide a complete audit trail for the alert condition checking and alert generation.

You can adjust pane sizes by dragging the horizontal bar separating the panes.

The top pane provides two views of the Alert Center configuration:

- **View by Alert Type** – This view displays all configured alerts and database systems, grouped by alert type and organized in a tree-like structure. Only database systems with audit trail monitoring jobs appear in this view.

  Tip: Because the audit monitoring jobs can be set up by different users using different DB Audit Management Consoles with different database profiles, you could potentially find alerts in this view for database systems you do not have configured in your console.

- **View by Database Server** – This view displays all configured alerts and database systems grouped by database system and organized in a tree-like structure. All database systems

whose profiles have been configured in the local DB Audit Management Console appear in this view regardless of whether any audit trail monitoring jobs have been setup for them.

Tip: In this view you can only see names of database systems whose profiles have been configured in your local DB Audit Management Console. Database systems and alerts for other systems are not displayed in this view.

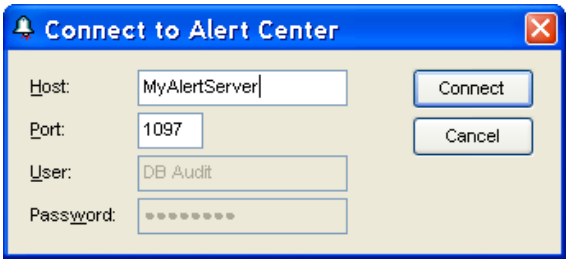
In both views you can use standard tree navigation techniques to collapse and expand groups of objects.

The following topics describe in detail how to use the Alert Center Remote Console to manage Alert Center settings, audit trail monitoring jobs and alert notifications.

Connecting to Alert Center

When launched, the Alert Center Remote Console automatically displays the **Connect to Alert Center** dialog you can use to specify required connection parameters and establish a new connection. The dialog automatically remembers the last used settings so you can simply click the **Connect** button on the dialog the next time you want to establish a connection to the Alert Center.

If you need to reconnect to a different Alert Center, use the **File/Connect to Alert Center** menu or click the **Connect** button on the console toolbar to break the current connection and invoke the **Connect to Alert Center** dialog.



The following parameters are required for a successful connection:

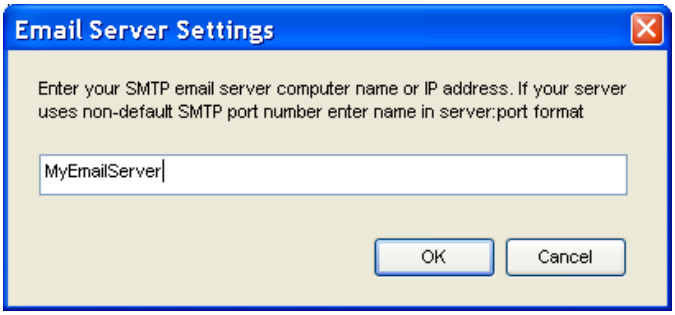
- **Host** – Enter the computer name or TCP/IP address of the computer running the Alert Center server.
- **Port** – Enter the Remote Control port number used by the Alert Center server. The default port is 1097. Do not change this number unless you have changed the Alert Center server and setup it to use a different port number.
- **User** – This parameter cannot be changed; it always defaults to **DB Audit** user.
- **Password** – This parameter cannot be changed ; it defaults to the password used internally by the **DB Audit** user.

After a successful connection has been established, the Alert Center Remote Console retrieves all configured database systems and their audit trail monitoring jobs and present this information in a graphical format.

Tip: You can use **File/Options** menu in the DB Audit Management Console to specify name and port for the default Alert Center server.

Configuring Alert Server Email Settings

You can use the **File/Set Email Server** menu or click the **Email Setup** button on the console toolbar to view and modify the Alert Center email settings. The Alert Center Remote Console will display **Email Server Settings** dialog as shown below.

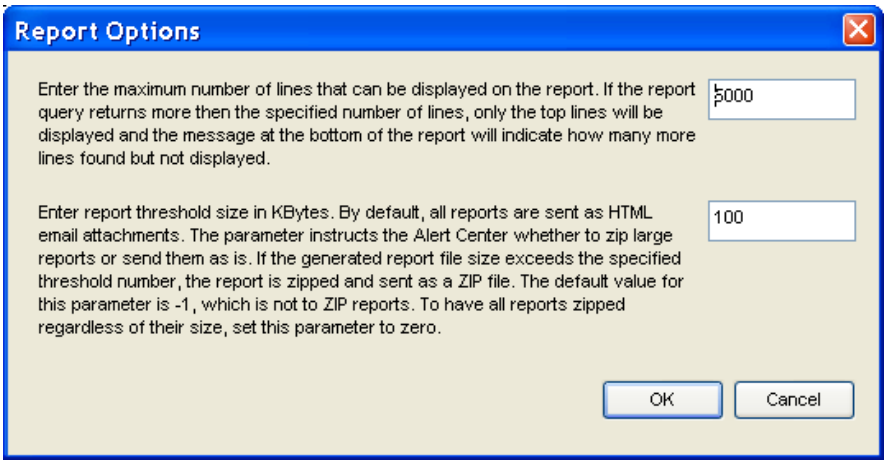


This dialog allows you to enter the name or TCP/IP address of your email server. Click the **OK** button to save your changes or click the **Cancel** button to close the dialog without saving changes.

Tip: By default, the Alert Center uses the default SMTP port number 25 assigned to SMTP protocol. If your email server is set to use a non-default SMTP port number, specify the email server name in **server:port** format. For example, *myserver:125* or *192.168.0.50:125*.

Configuring Report Generation Options

You can use the **File/Set Report Options** menu or click the **Options** button on the console toolbar to view and modify the Alert Center report generation settings. The Alert Center Remote Console will display **Report Options** dialog as shown below.



Tip: The complete description and instructions for changing these options are provided directly on

the screen.

Alert Management

The Alert Center Remote Console provides complete set of functions for managing audit trail monitoring jobs and alerts.

To simplify alert management, the Alert Center Remote Console supports two modes of operations on alerts:

- Operations on individual alerts such as create, delete, disable and so on. This mode provides the most flexibility and full access to all supported alert types, including creation of new user-defined alert types. See Creating Alerts topic in this chapter for more details.

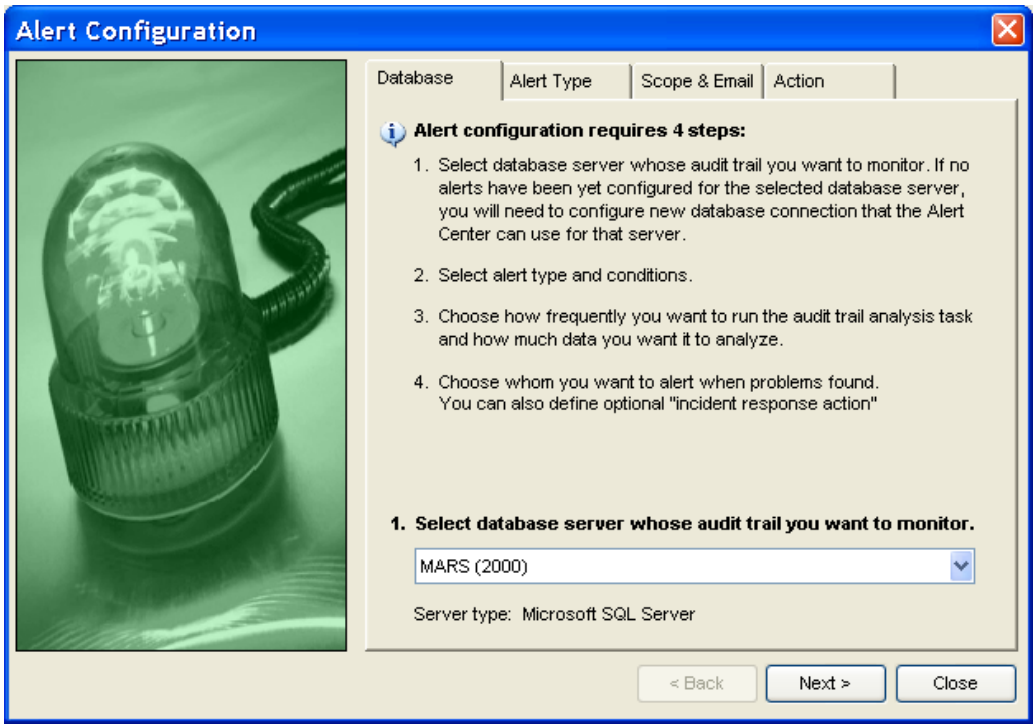
- Batch operations affecting multiple alerts and database systems. This mode is the most convenient way to modify multiple alerts for one or more configured database systems. This method cannot be used to create certain alert types or to create new user-defined alert types.

The following topics describe in details all supported alert management functions and modes..

Creating Alerts

To create a new alert:

1. Choose source of the report data – In the Alert Center Remote Console interface select either of the following items in the system tree:
   - Alerts and Reports from Central Repository Server
   - Alerts and Reports from Audited Database Server
2. Click the **Alerts/New Alert** menu, press the **CTRL+N** key combination, or click the **New** button on the Alert Center Remote Console toolbar. The Alert Center Remote Console will display the **Alert Configuration** dialog which can be used to create new or modify existing alerts. Below is an example screenshot of the **Alert Configuration** dialog.



The dialog consists of a tabbed interface which can be navigated using the tabs displayed on the top of the dialog screen or by using the **Next** and **Back** buttons at the bottom of the dialog.

To set up a new alert you must complete the following four configuration steps:

3. In the **Database Profiles** drop-down list, select the database server whose audit trail you want to monitor. Click the **Next** button to advance to the next step.

4. In the **Alert Types** drop-down list, select the required alert type and then enter the required alert condition. Note that different conditional parameters are applicable to different alert types. The Alert Center Remote Console enables and allows entering only those parameters that are applicable to the selected alert type. When finished, click the **Next** button to advance to the next step.

5. In the **Check Frequency** drop-down list, select how frequently you want to run the audit trail monitoring job. If you don't see an appropriate value, select the **[custom interval]** option and enter a frequency value in the **Custom Frequency** input field. The frequency value must be entered in minutes.

   In the **Scope** drop-down list, select the length of the time period for which you want the audit trail monitoring job to run. If you do not see an appropriate value, select the **[custom interval]** option and enter an appropriate value in the **Custom Scope** input field. The scope value must be entered in minutes.

6. In the **Recipient's Email** input field, enter the email address of the person or distribution group to whom you want to alert to be sent.

   If your email server requires user authentication, enter a user name and password in the **Sender's Email/Name** and **Sender's Password** fields.

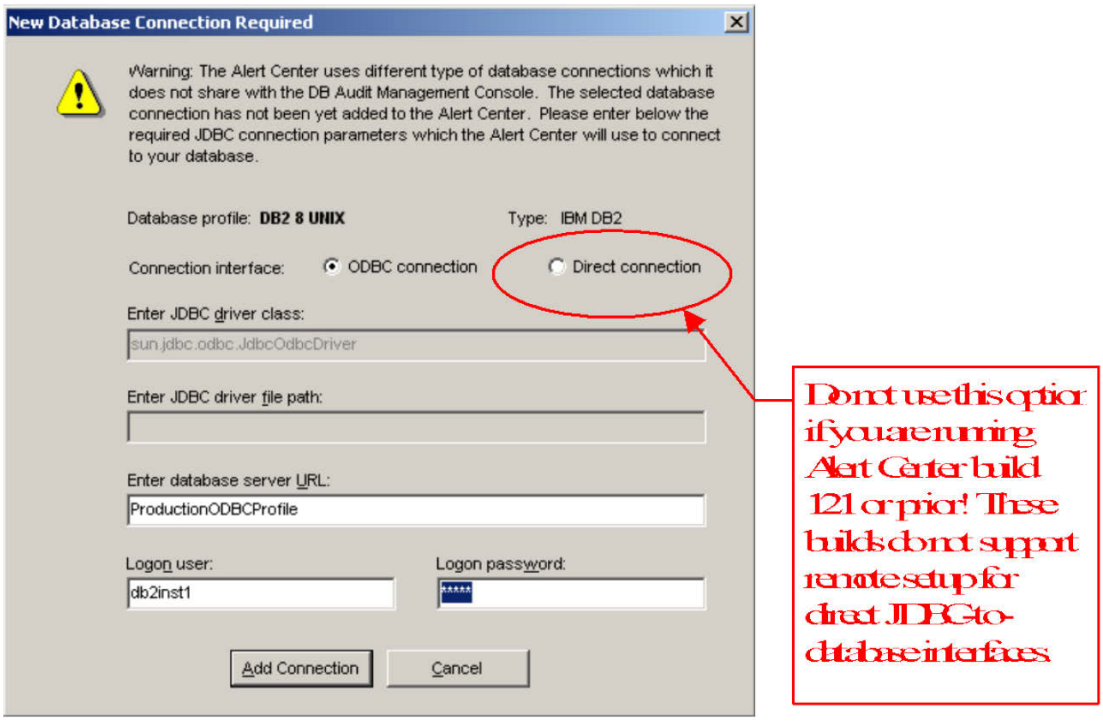
   Click the **Finish** button to validate the alert parameters and then close the dialog.

   If the new alert refers to a database server whose connection has not been yet configured in the Alert Center, the Alert Center Remote Console automatically displays

the **New Database Connection Required** dialog. This dialog is used to enter database connection parameters for the Alert Center server.

Notes: Do not confuse database profiles used by the DB Audit Management Console with database profiles used by the Alert Center server. These are different profiles used by different programs that could be running on different computers using different connection methods.

The **New Database Connection Required** dialog allows you to specify connection interface properties for the Alert Center server only!



On the **New Database Connection Required** dialog:

1. Select **Direct Connection** or **ODBC connection** option as the Connection interface.

2. If you selected **Direct connection** type, enter placeholder values for the database server name and port created for you in the **Database Server URL** input box.

   Note: Make sure you do not leave brackets indicating positions of placeholder; for example, if the placeholder appears as [SERVER NAME], make sure you replace the entire placeholder including the [ ] brackets with the actual server name.

   If you selected **ODBC connection** type, enter the name of an existing ODBC profile into the **Database Server URL** input box.

   Note: Please make sure you enter an ODBC profile name available on the computer running the Alert Center, not one on the local computer from which you are remotely configuring the new alert and new database connection.

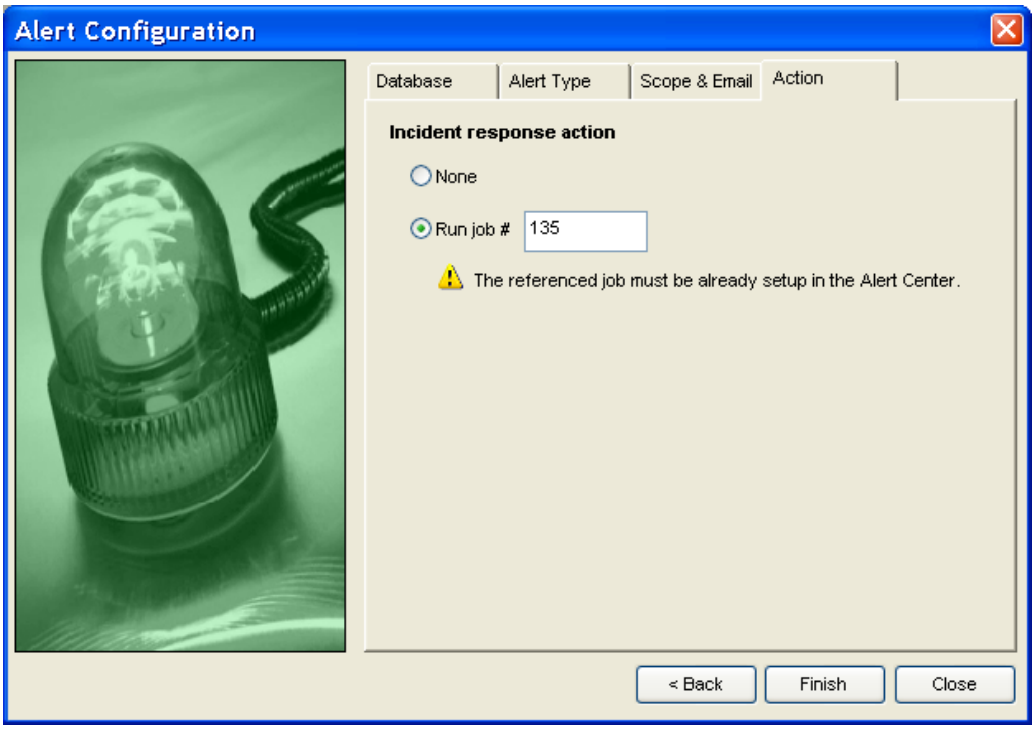
3. If you are using a non-trusted connection, enter the database server logon and password. The specified logon must have permissions to execute SELECT type SQL queries on the system audit tables. For information on the system audit trail

location and table names, see Configuring System Audit Options topics in CHAPTER 3.

4. Click the **Add Connection** button to add the new connection.

Tip: Operations 2 through 4 are only required once for each database connection. Once you have created the connection, you can use it to add additional alerts and modify existing alerts.

Optionally you can assign an incident response action to the created alert. On the **Action** tab page, select the **Run Job** option and then enter the numeric ID of the required action job.



Tip: Action jobs can be shared by multiple alert monitoring jobs.

For information on how to create action jobs, see the Creating Incident Response Jobs topic in this chapter.

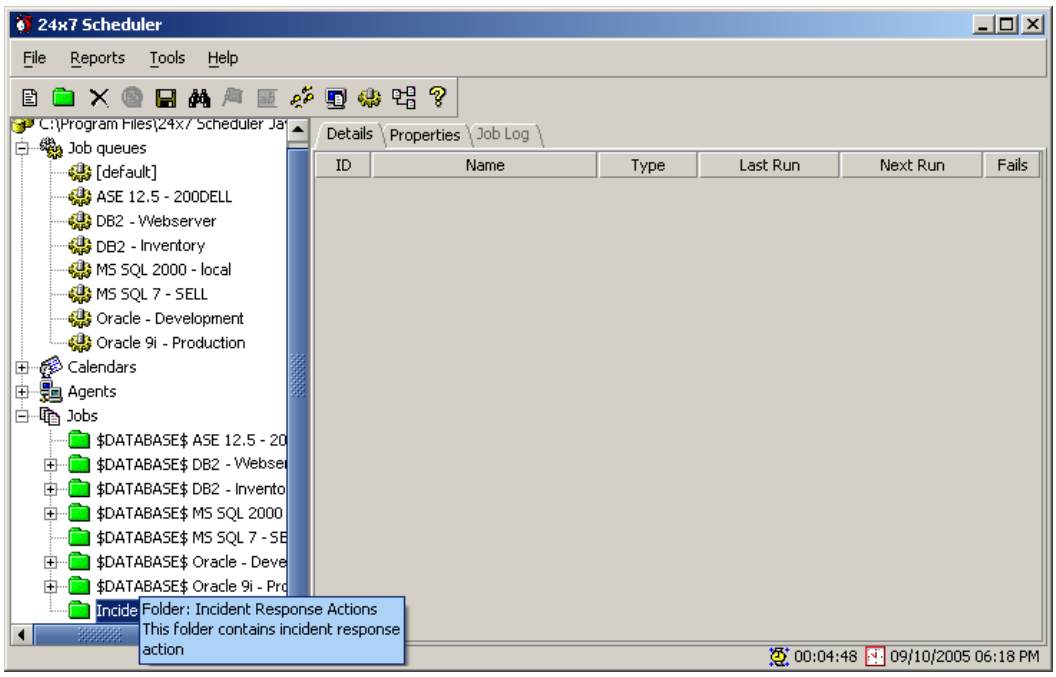
Creating Incident Response Jobs

You can create incident response jobs using either the 24x7 Scheduler graphical interface or the 24x7 Web-based Management Console. Use the graphical interface when you have direct access to the server running the Alert Center and the scheduler is running in graphical mode. Use the web-based console when you don't have access to the Alert Center server or when the scheduler is running in non-graphical mode.

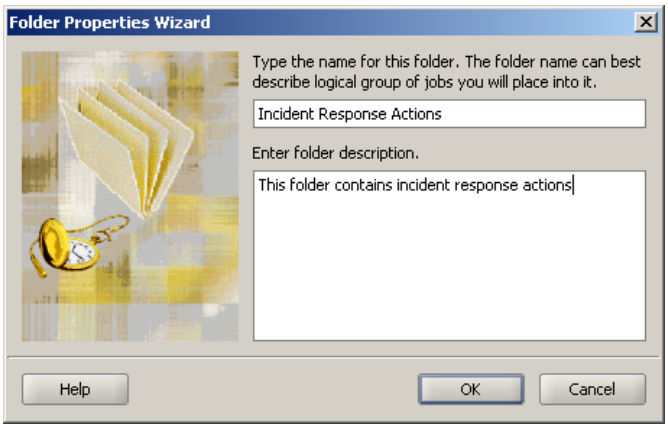
The following example demonstrates how to create a new incident response job using the graphical interface. This example assumes the 24x7 Scheduler is running in graphical mode.

1. **NOTE:** This step must be performed only once when you create the first incident response job. If you have already created a folder, select the existing folder then go on to step 2.

On the 24x7 Scheduler toolbar, create an incident response folder by clicking the **New Folder** button . 'Alternatively, you can click **File/New/Folder** menu.



The **Folder Properties Wizard** dialog will appear. Enter a descriptive folder name and folder description; for example, "*Incident Response Actions*" and "*This folder contains incident response action jobs*."



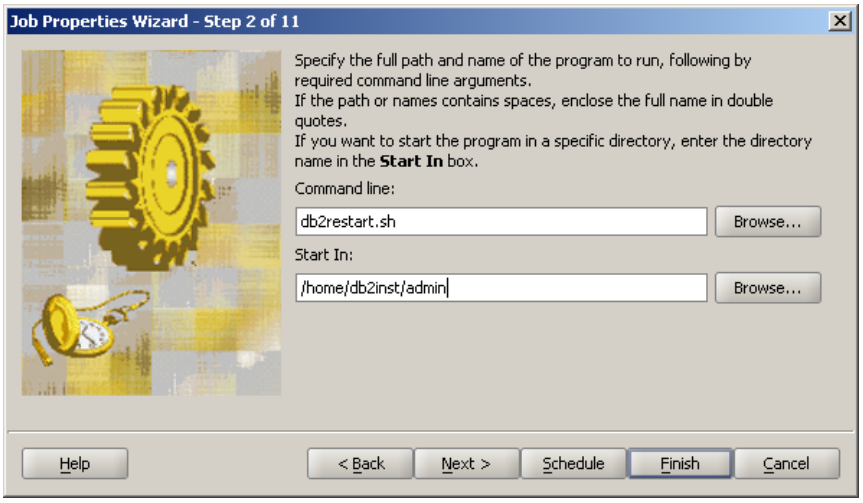
Click the **OK** button to close the **Folder Properties Wizard** dialog.

2. On the 24x7 Scheduler toolbar, click the **New Folder** button . 'Alternatively, you can click the **File > New > Job** menu or press CTRL+N keyboard shortcut.

   The **Job Properties Wizard** dialog will appear. In the job name box, enter a descriptive job name; for example, "*Restart DB2 database.*" Choose the appropriate option for the job type. In this example we will use the **Run program** option. For detailed constructions on how to use other job types, see your 24x7 Scheduler User's Guide.

Click the **Next** button to advance to the next step.

3. Enter the required job **Command Line** and **Start In** directory name. For example, enter name of shell file *db2restart.sh* and */home/db2inst/admin* for the command line and directory name.



Create the specified shell script file if it does not yet exist. You can use any available text editor or scripting utility for that purpose. The script must be created on the computer where the job will run. For example, enter the following two lines as the script.

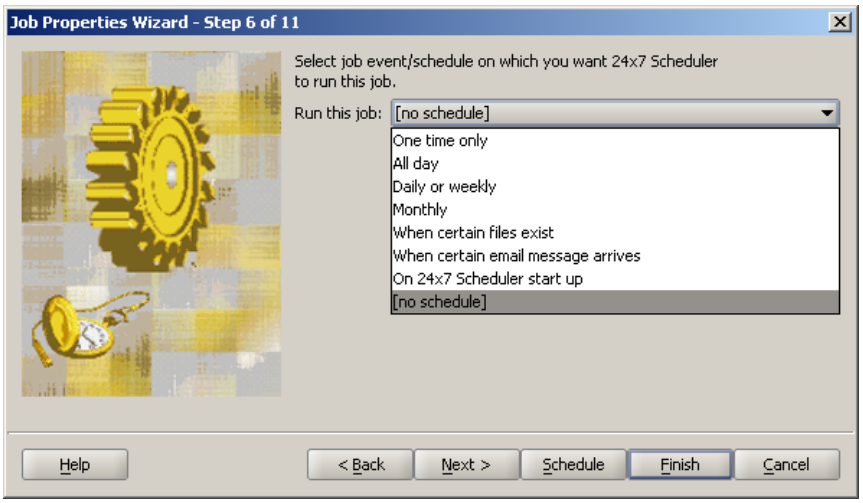
```
db2stop /FORCE
db2start
```

If the Alert Center and the database server run on the same computer, the job needs to be run locally. If the database server resides on another computer, this example job needs to be run on the database server computer. In that case, you must first install 24x7 Remote Agents on database server computer and in the 24x7 Scheduler settings, define a named remote agent profile for that agent. You can then select the profile name in the job properties.

For simplicity, this example assumes that the Alert Center is set up on the same computer so that the job needs to be run locally.

Click the **Schedule** button to advance to the job-scheduling step.

4. In the **Run This Job** drop-down schedule list, select **[No schedule]** item. This is because the action job is an event-based job. It is started automatically by audit trail monitoring jobs that you link to this action job.



Click the **Finish** button the **Job Properties Wizard** dialog. On the 24x7 Scheduler toolbar click the **Save** button to save all changes. Alternatively, you can click **File > Save** menu or press CTRL+S keyboard shortcut.

Tip: 24x7 Scheduler is an advanced enterprise job scheduling system that supports a vast array of functions and job scheduling methods. The complete description of all available features, functions and methods is provided in 24x7 Scheduler User's Guide.

Creating Custom Alerts

To create a new user-defined alert, follow the steps described in the Creating Alerts topic in this chapter. In step two, when prompted for the alert type, choose the **[Custom Alert]** item in the **Alert Types** drop-down list. The **Custom SQL** edit box will appear below the drop-down. Type or paste in your SQL statement. The SQL statement must satisfy the following rules:

- You must enter either a valid SQL SELECT statement supported by your database system or a valid command calling a stored procedure. The stored procedure must return either a result set or a batch SQL block that returns a result set.

  Oracle: Only a single SELECT statement can be used. The SELECT can have any number of subqueries, joins, etc…

  DB2: You can use either a SELECT statement or a CALL statement to call a stored procedure. If you use a CALL statement, the called stored procedure must open a cursor and leave it open when the procedure completes.

  SQL Server, ASE, ASA: You can use any of the methods described in this paragraph. If you create a temporary table within a called procedure or T-SQL block, make sure to drop that table before your SQL command completes.

- The result set must return no records if no problems have been found. If problems have been found, it must return one or more records identifying the problems. It is generally recommended for performance reasons that you design the SQL statement to return no more than a single record in the output result set.

Example 1 (DB2):

```
SELECT count(*)
FROM db_audit.sys_audit_trail
WHERE category = 'VALIDATE'
    AND event IN ('AUTHENTICATION', 'CHECK_GROUP_MEMBERSHIP')
    AND status < 0
    AND eventtime BETWEEN current timestamp - 30 minutes
                      AND current timestamp
HAVING count(*) > 1000
```

Example 2 (Oracle):

```
SELECT *
FROM (SELECT count(1) AS counter
    FROM sys.dba_audit_trail
    WHERE action_name = 'LOGON'
       AND returncode != 0
       AND timestamp BETWEEN sysdate - 1/24*30 AND sysdate
    )
WHERE counter > 1000
```

- The SQL statement must not create any locks in the database that could disrupt or otherwise affect the auditing processes. Failure to follow this rule may cause the auditing or even the entire database server to hang.

- In addition to these mandatory rules, you should optimize your custom alert's SQL statement to run as fast as possible so that it does not cause performance problems for the database and for the Alert Center server.

Modifying Alerts

To modify parameters of an existing alert, select it in the alert tree displayed inside the Alert Center Remote Console. To select the alert, double-click the alert name, click the **Alerts > Edit** menu, press the **CTRL+E** hot key, or click the **Edit**  button on the Alert Center Remote Console toolbar. The Alert Center Remote Console will display **Alert Configuration** dialog, which can be used to modify the existing alert. This is the same dialog that you use to create new alerts. You can use it to modify any alert parameters except the alert type and the assigned database connection.

For more information on entering or modifying alert parameters, see Creating Alerts topic in this chapter.

Deleting Alerts

To delete of an existing alert, select it in the alert tree displayed inside the Alert Center Remote Console. You can then use either of the following: click the **Alerts > Delete** menu, press the **CTRL+D** hot key, or click the **Delete**  button on the Alert Center Remote Console toolbar.

Disabling and Enabling Alerts

Use the disable option to temporarily disable existing audit trail monitoring jobs without permanently deleting alert definitions from the Alert Center.

Use the enable option to enable previously disabled alerts.

To disable or enable an existing alert, select it in the alert tree displayed inside the Alert Center Remote Console. You can then use any of the following: click the **Alerts > Disable > Enable** menu, press **CTRL+B** hot key, or click the **Disable**  button on the Alert Center Remote Console toolbar.

Manually Running Audit Trail Monitoring Jobs

To verify that an alert is working correctly, you can manually trigger the run of its audit trail monitoring job.

To trigger run of an existing audit trail monitoring job, select the desired alert in the alert tree, which is displayed inside the Alert Center Remote Console. You can then use any of the following: click the **Alerts > Run Now** menu, press the **CTRL+R** hot key, or click the **Run Now**  button on the Alert Center Remote Console toolbar. The Alert Center server will run the job and post job status messages to the alert job log. The Alert Center Remote Console will wait for the job to complete and then automatically refresh the alert log view displayed inside the bottom pane of the Alert Center Remote Console screen.

Reviewing Alert Logs

The Alert Center Remote Console bottom pane has two functions: it can display processing messages for individual audit trail monitoring jobs, and it can display processing messages for the entire Alert Center server, including all alerts and audit trail monitoring jobs.

To see the alert log for an individual alert: select that alert in the alert tree displayed inside the top pane of the Alert Center Remote Console.

To see the alert log for all alerts: select the top **Alerts** option displayed in the root of the alerts tree.

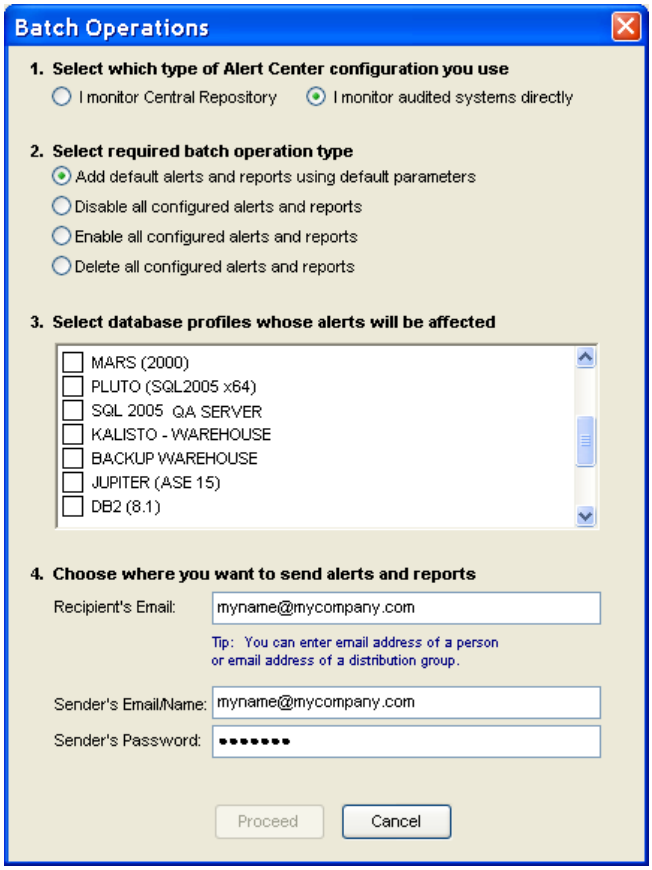
To refresh the current view: click the **View > Refresh** menu, press the **F5** hot key, or click the **Refresh**  button on the Alert Center Remote Console toolbar.

Performing Batch Operations on Alerts and Reports

To simplify alert management and system maintenance tasks, the Alert Center Remote Console provides support for batch operations on alerts. You can these batch operations to perform the following tasks:

- Create multiple alerts using default alert configuration parameters and add these alerts to one or more configured database systems.

- Disable all configured alerts for one or more configured database systems.
- Enable all previously disabled alerts for one or more configured database systems.
- Delete all configured alerts for one or more configured database systems.



To perform any of these batch operations, click the **Alerts > Batch Operations** menu or click the **Batch** button on the Alert Center Remote Console toolbar. This will display the **Batch Operations** dialog.

The **Batch Operations** dialog requires you to complete the following steps:

1. Select operation type.

2. Select one or more database profiles whose alerts will be affected by the batch operation.

3. For an “add alerts” operation, specify email settings:

   In the **Recipient's Email** input field, enter the email address of the person or distribution list to whom you want the alert to be sent.

   If your email server requires user authentication, fill in **Sender's Email/Name** and **Sender's Password** fields.

   Click the **Proceed** button to perform the selected operation, or click the **Cancel** button to cancel and close the dialog.

Supported Alert Types

The following topics describe supported alert types, how they work and their default parameters.

Some alert types analyze data in the system audit trail checking for specific events, while other alert types do not use audit trail data at all, these alert types check various database performance characteristics.

For the alert type descriptions below, minutes are used as the unit of measurement for the **Scope** parameter. For example, when we say that alert scope is the last 60 minutes of data, we mean that the associated audit trail monitoring job only checks audit records having timestamps within the previous 60 minutes. The actual number of records or volume of data processed within the specified scope period could vary in different systems and at different times; it greatly depends on what auditing operations have been enabled and how busy the database system is.

Intrusion Detection

Lots of Failed Connection Attempts in a Short Period of Time

This alert checks the system audit trail for “failed” logon events. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified **Scope** parameter. The **Fequency** parameter controls how frequently to run the monitoring job. The **Threshold** parameter defines the conditions under which an alert will be generated.

Requirements: The system auditing must be installed and enabled. The logon auditing options must be enabled.

ASA: This alert type is not supported

DB2: This alert type is currently supported only by DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes
Threshold Value: at least 100 failed logons occurred during the specified analysis time

Lots of Connection Attempts from the Same Terminal Using Different User Names

This alert checks the system audit trail for multiple logon attempts from the same network terminal or workstations using different logon names. The alert does not care if the recorded logons succeeded or failed. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified threshold parameter defines when to generate an alert.

Requirements: The system auditing must be installed and enabled. The logon auditing options must be enabled.

ASA: This alert type is not supported

DB2: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes
Threshold Value: at least 100 logon attempts with different logon names occurred during the specified analysis time

Connection Attempts From Terminals Not in Your Network Domain

This alert checks the system audit trail for logons attempted from network terminals or workstations whose users logged on to a network domain different from the user-specified domain. The alert does not care if the recorded logons succeeded or failed. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified threshold parameter defines when to generate an alert. The user-specified domain parameter defines the baseline domain name. Normally you should specify your primary domain name or the domain your database server belongs to.

Requirements: The system auditing must be installed and enabled. The logon auditing options must be enabled.

ASA: This alert type is not supported

DB2: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes
Threshold Value: at least 10 logon attempts from other domain occurred during the specified analysis time
Domain name: domain name for the workstation from where the alert definition has been created

Denial of Service Attack

Lots of Connection Attempts in a Short Period of Time

This alert checks the system audit trail for a large number of logons attempted during a short period of time. The alert does not care if the recorded logons succeeded or failed. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified threshold parameter defines when to generate an alert.

Requirements: The system auditing must be installed and enabled. The logon auditing options must be enabled.

ASA: This alert type is not supported

DB2: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes
Threshold Value: at least 1000 logon attempts occurred during the specified analysis time

Slow System Response Executing Simple Queries

This alert checks performance of simple queries executed periodically by the associated database monitoring job. The actual SQL statements executed queries varies for different systems. The following queries are used:

- **Oracle**: `SELECT 2.5*31/64.2, sysdate FROM dual`
- **SQL Server, ASE, ASA**: `SELECT 2.5*31/64.2, GetDate()`
- **DB2 (UDB for Linux, Unix and Windows, UDB for zSeries)**: `SELECT 2.5*31/64.2, current timestamp FROM sysibm.sysdummy1`
- **DB2 for iSeries:** This alert type is not currently supported.

The associated database job attempts to run the query three times with a 5-second pause between runs. It then calculates average query execution time. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified threshold parameter defines when to generate an alert.

Requirements:

- **DB2**: This alert type is currently not supported with DB2 for Series

Default parameters:
Frequency: run monitoring job every 10 minutes
Threshold Value: average query execution time is over 5 seconds

Consecutive Connection Failures

This alert checks database connectivity issues. The associated database monitoring job periodically attempts to connect to the database server and, if a connection fails, makes several more consecutive attempts, checking whether the connectivity problem is persistent or random. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified threshold parameter defines how many times to retry connecting to the database before generating an alert.

Requirements: None

Default parameters:
Frequency: run monitoring job every 10 minutes
Threshold Value: retry 3 times

Connection Handshake Taking Long Time

This alert checks database connectivity issues. The associated database monitoring job periodically attempts to connect to the database server and, if the connection attempt takes a long time, makes three more consecutive attempts with a 5-second pause between attempts, checking whether the connectivity problem is persistent or random. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified threshold parameter defines “long” connection time, which in turn defines when to generate an alert.

Requirements: None

Default parameters:

Frequency: run monitoring job every 10 minutes
Threshold Value: connection handshake takes 10 or more seconds

Unauthorized Access Attempts

Access Denial Events

This alert checks the system audit trail for “access denied” events. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job.

Requirements: The system auditing must be installed and enabled. The “failed object access” and “failed operation attempts” auditing options must be enabled.

 **ASA**: This alert type is not supported

 **DB2**: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes

Usage Notes: The generated alert email includes a detailed Access Denied HTML report, which is sent as an email attachment. The report lists detected violations including who, when, what and where descriptors. If the number of violations is high, the report is limited to the first 100 events.

Database Errors

Excessive Number of Database Errors in a Short Period of Time

This alert checks the system audit trail for significant number of “database error” events or other events that have completion statuses indicating various database processing errors. Note that failed logons do not fall into this category. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified threshold parameter defines when to generate an alert.

Requirements: The system auditing must be installed and enabled. If the database system and DB Audit do not support the separate auditing option for “database errors” events, then you must have enabled all audit options for “object access” and “statement execution”. If the database system and DB Audit do support the separate auditing option for “database errors,” it is sufficient to have only this audit option enabled.

 **ASA**: This alert type is not supported

 **DB2**: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 30 minutes of audit trail data
Frequency: run monitoring job every 15 minutes
Threshold: At least 100 errors occurred during the specified analysis time

Usage Notes: The generated alert email includes a detailed Database Errors HTML report, which is sent as an email attachment. The report lists detected errors including who, when, what and where

descriptors. If the number of error is high, the report is limited to the first 100 events.

Important Notes:

- Different database systems define database errors differently. For instance, in Oracle and SQL Server, UPDATE operations that find no data to update are not considered as errors, while in DB2 they generate a “no data found” warning and are recorded in the audit trail with no zero statuses indicating an error or warning condition. On the other hand, in Oracle, a SELECT operation that fails due to insufficient user permissions will be recorded as an error while in DB2 it will not be recorded at all. DB2 does record “access denied” events as errors.

- “Failed logon” events are not analyzed by this alert.

- Database transactions rolled back as a result of ROLLBACK statement execution do not automatically indicate an error condition. Rollback processing can be caused by certain business logic implementation embedded in user programs.

Certain Types of Database Errors

This alert checks the system audit trail for specific “database error” events or other events that have specific completion statuses. Note that failed logons do not fall into this category. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified error-list parameter defines which specific errors to monitor, which effectively defines when to generate an alert.

Requirements: System auditing must be installed and enabled. If the database system and DB Audit do not support the separate auditing option for “database errors” events, you enable all audit options for “object access” and “statement execution”. If the database system and DB Audit do support the separate auditing option for “database errors,” it is sufficient to have only this audit option enabled.

ASA: This alert type is not supported

DB2: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 30 minutes of audit trail data
Frequency: run monitoring job every 15 minutes

Usage Notes:
Error numbers must be specified as a comma-separated list.

The generated alert email includes detailed Database Errors HTML report, which is sent as an email attachment. The report lists detected errors including who, when, what and where descriptors. If the number of error is high, the report is limited to the first 100 events.

Suspicious Activities

Excessive Number of Certain Queries in a Short Period of Time

This alert checks the system audit trail for SQL queries containing specific text. The associated audit

trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified substring-list parameter defines which specific query text to monitor. The user-specified threshold parameter defines when to generate an alert.

Requirements: System auditing must be installed and enabled. The "audit SQL" options must be enabled.

Oracle: This alert type is limited to checking operation type names such as SELECT, DROP SEQUENCE, TRUNCATE, CREATE USER, and so on.

ASA: This alert type is not supported

DB2: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes
Threshold Value: at least one query containing the specified text must have occurred during the specified analysis time

Usage Notes:
Search substrings must be specified as a comma-separated list. The search substring is not case-sensitive.

The generated alert email includes a detailed SQL Queries HTML report, which is sent as an email attachment. The report lists found SQL Queries including who, when, what and where descriptors. If the number of queries is high, the report is limited to the first 100 events.

Access to Certain Tables After Regular Business Hours

This alert checks the system audit trail for access to specific tables and views that occur after regular business hours. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified object-list parameter defines which specific objects to monitor. The user-specified business day start and end time parameters define what is considered as a business day.

Requirements: System auditing must be installed and enabled. The "object access" audit options must be enabled.

ASA: This alert type is not supported

DB2: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes

Usage Notes:
Object names must be specified as a comma-separated list. Because the performed analysis is not case-sensitive names can be entered in lower, upper or mixed case. If you want to limit the monitoring to specific schemas, specify object names in SCHEMA.OBJECT format. If the database server supports multiple databases and you want to limit the search to specific databases, specify object names in DATABASE.SCHEMA.OBJECT format.

The generated alert email includes a detailed Table Access HTML report, which is sent as an email attachment. The report lists found objects including who, when, what and where descriptors. If the number of queries is high, the report is limited to the first 100 events.

Data Changes in Certain Tables

This alert checks the system audit trail for data change events in specific tables. In particular it checks for DELETE, UPDATE, INSERT and TRUNCATE operations. The associated audit trail monitoring job analyzes recent audit trail data as defined by the user-specified scope parameter. The user-specified frequency parameter controls how frequently to run the monitoring job. The user-specified object-list parameter defines which tables to monitor.

Requirements: System auditing must be installed and enabled. Either the "object access" audit options must be enabled or "statement" audit options for auditing of DELETE, INSERT, UPDATE and TRUNCATE operations must be enabled.

 **ASA**: This alert type is not supported

 **DB2**: This alert type is currently supported only with DB2 UDB for Linux, Unix and Windows

Default parameters:
Scope: last 60 minutes of audit trail data
Frequency: run monitoring job every 30 minutes

Usage Notes:
Object names must be specified as a comma-separated list. Because the performed analysis is not case-sensitive names can be entered in lower, upper or mixed case.. If you want to limit the monitoring to specific schemas, specify object names in SCHEMA.OBJECT format. If the database server supports multiple databases and you want to limit the monitoring to specific databases, specify object names in DATABASE.SCHEMA.OBJECT format.

The generated alert email includes a detailed Table Access HTML report, which is sent as an email attachment. The report lists found objects including who, when, what and where descriptors. If the number of queries is high, the report is limited to the first 100 events.

Custom Alerts

The Alert Center gives you the flexibility to define custom alert types. You can use virtually any valid SQL commands in the alert query and can set the query to run as often as needed. The Alert Center will run your queries and analyze their results. Should your query return any results considered as alert conditions, the Alert Center will automatically generate a new alert and notify you via email of any problems found. For more information on how to create custom alerts, see Creating Custom Alerts topic.

.

CHAPTER 7: Reports

Report Types

DB Audit supports two types of pre-built reports:

- Interactive Graphical Reports – these reports are accessible via the DB Audit Management Console or the Report Viewer utility.
- Scheduled Reports – these reports are non-interactive; they are scheduled and run using the Alert Center and are automatically delivered by email.

DB Audit allows you to create custom reports:

If DB Audit doesn't already have the pre-built report you need, you can easily create your own. For example, you can create a custom report that color-codes different types of activity records or highlights in yellow any questionable applications that a user has run. Custom reports can be created using DB Audit tools as well as using any other reporting tools of your choice. The audit data is provided to you in a convenient structured format stored in audit trail tables. These tables can be queried just like any other relational database tables. See CHAPTER 3: System Auditing and CHAPTER 4: Data Change Auditing for more information on audit trail tables, their names and locations. For custom report examples see Custom Reports topic in CHAPTER 7.

Interactive Graphical Reports

DB Audit Expert provides over 50 pre-built interactive reports that can be used for:

- Tracking data changes in the database – see Data-Change Audit Reports topic for more information.
- Displaying enabled audit options and settings – see Enabled System Audits topic for more information.
- Analyzing system security and identifying various database security violations – see System Audit and Security Reports topic for more information.
- Checking and ensuring SOX compliance reports – see Compliance Reports topic for more information.
- Analyzing user behavior analysis reports – see Behavioral Analysis Reports topic for more information.
- Performing statistical analysis of the system audit trail data and discovering anomalies – see Statistical Reports topic for more information.
- Taking a quick glance at the current date events – see DB Audit Start Page topic for more information.

Scheduled Reports

You can use the Alert Center Remote Console to schedule automatic reports delivered directly into

your email inbox. Reports can be scheduled to run at a time and frequency of your choosing.

If you run a central repository system, you can also use the Central Repository Deployment Tools to add or delete pre-configured reports to/from central repository

The pre-built reports can be used mostly for analyzing system security and identifying various database security violations. You can also create your custom reports.

Read Scheduled System Audit Reports topic in this chapter for descriptions of pre-built reports

Data-Change Audit Reports

Enabled Data Change Audits Report

This report returns the complete list of tables currently setup for data-change auditing as well as the audit settings for each table. To run the report, select the **Reports > Data Change Audit Configuration > Enabled Data change Audits** menu.

The following columns are displayed on the report:

1.  **ASE, SQL Server: Database** – Name of the database containing the audited table.
2. **Schema –** Name of the database schema the audited table belongs to.
3. **Table Name –** Name of the audited table.
4. **Table Alias –** User-oriented descriptive table name if available. For more info, see "Setting Table and Column Aliases" topic in CHAPTER 7.
5. **Audit Ins –** Audit state for data INSERT operations. A marked check box in this column indicates that INSERT operations are captured and recorded in the audit trail.
6. **Audit Del –** Audit state for data DELETE operations. A marked check box in this column indicates that DELETE operations are captured and recorded in the audit trail.
7. **Audit Upd –** Audit state for data UPDATE operations. A marked check box in this column indicates that UPDATE operations are captured and recorded in the audit trail.
8. **Audit Users –** Brief definition of the user-level audit filter.
9. **Email Alert –** State of the data-change alerts. A marked check box in this column indicates that data change alerts are enabled.
10. **Email Alert Recipients –** Email addresses selected for receiving of data-change email alerts.

User-level Audit Filters Report

This report returns a list of tables currently set up for data-change auditing whose audit configuration includes user-level AUDIT and NO AUDIT filters. To run the report, select the **Reports > Data Change Audit Configuration > User-level Audit Filters** menu.

The following columns are displayed on the report:

1. **ASE, SQL Server: Database** – Name of the database containing the audited table.
2. **Schema –** Name of the database schema the audited table belongs to.
3. **Table Name –** Name of the audited table.
4. **User Name –** Name of the user added to the audit filter.
5. **Audit Rule –** One of the following: "Audit" or "Do not Audit"

Application-level Audit Filters Report

This report returns a list of tables currently set up for data-change auditing whose audit configuration includes application-level AUDIT and NO AUDIT filters. To run the report, select the **Reports > Data Change Audit Configuration > Application-level Audit Filters** menu.

The following columns are displayed on the report:

1. **ASE, SQL Server: Database** – Name of the database containing the audited table.
2. **Schema –** Name of the database schema the audited table belongs to.
3. **Table Name –** Name of the audited table.
4. **Application Name –** Name of the application added to the audit filter.
5. **Audit Rule –** One of the following "Audit" or "Do not Audit"

Audit Trail Table Detail Report

This report returns detailed information from the data audit trail. To run the report, select the **Reports > Data Change Audit Trail > Audit Trail By Table** menu. The **Select Audited Table** dialog will appear. Choose the desired table and click the **OK** button. The **Specify Report Filter** dialog will then appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the database audit trail tables.

The following columns are displayed on the report:

1. **Timestamp** – The date and time of the change.
2. **Program Name –** The name of the program used to make changes.
3. **Terminal –** Network name of the terminal or user workstation where these changes were made.
4. **Oracle, ASA, DB2: Network User –** Network name of the user who made the change. If the database system uses trusted connections, this value will match the value in the **Database User** column.
5. **ASE, SQL Server: Login Name –** Login name of the user who made the change. If the database system uses trusted connections, this value will match the user's network name.
6. **Database User –** Name of the database user who made the change.
7. **Operation –** Type of database operation that caused the change; i.e., one of the following: INSERT, UPDATE or DELETE.

8. **Value Type** – Displays one of the following values types:

   OLD – this value is always used for DELETE operations. For UPDATE operations, this value indicates that the displayed record contains values before the change occurred.

   NEW – this value is always used for INSERT operations. For UPDATE operations, this value indicates that the displayed record contains values after the change occurred.

9. The remaining columns are the columns selected for auditing.

Note:
The audit trail reports cannot display columns with user-defined data types that do not map directly to the standard ANSI compliant data types. In addition to user-defined data types, the following data types native to that particular DBMS also cannot be displayed:

Oracle: BLOB, BFILE, CLOB, RAW, LONG RAW, object, VARRAY, nested table, and REF columns.

SQL Server, ASE, ASA: IMAGE and all data types that cannot be converted to VARCHAR using the built-in *convert* function.

DB2: BLOB, CLOB, DBLOB and derivative types.

Audit Trail Table Summary Report

This report returns data from the data audit trail. To run the report, select the **Reports > Data Change Audit Trail > Audit Trail Summary By Table** menu. The **Select Audited Table** dialog will appear. Choose the desired table and click the **OK** button. The **Specify Report Filter** dialog will then appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the database audit trail tables.

Year, month, user, program, terminal and type of the operations group the report data. The following columns are displayed on the report:

1. **Year/Month** – The year and month when changes were made.
2. **Program Name –** The name of the program used to make changes.
3. **Terminal –** Network name of the terminal or user workstation where the changes were made from.
4. **Oracle, ASA, DB2: Network User –** Network name of the user who made these changes. If the database system uses trusted connections, this value will match the value in the next **Database User** column.
5. **ASE, SQL Server: Login Name –** Login name of the user who made these changes. If the database system uses trusted connections, this value will match the user's network name.
6. **Database User –** Name of the database user who made these changes.
7. **Operation –** Type of the database operation that caused the changes; i.e., one of the following: INSERT, UPDATE or DELETE.
8. **Rows Affected –** Total number of records that were changed by the user or program during the reported year and month specified in other report columns

Audit Summary for All Tables Report

This report returns data from the data audit trail. To run the report, select the **Reports > Data Change Audit Trail > Audit Trail Summary By Table** menu. The **Specify Report Filter** dialog will appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the database audit trail tables.

The report data is grouped by audited table name, program, terminal and user. The following columns are displayed on the report:

1.  **Oracle, ASA, DB2: Table Name** – The full table name including schema name.
    **ASE, SQL Server: Table Name** – The full table name including database and schema names.
2. **Program Name –** The name of the program used to make these changes.
3. **Terminal –** Network name of the terminal or user workstation where these changes were made from.
4.  **Oracle, ASA, DB2: Network User –** Network name of the user who made changes. If the database system uses trusted connections this value should match the value in the next **Database User** column.
5.  **ASE, SQL Server: Login Name –** Login name of the user who made changes. If the database system uses trusted connections, this value should match user's network name.
6. **Database User –** Name of the database user who made changes.
7. **Rows Inserted –** Total number of records that were added to the audited table by the user or program specified in other report columns.
8. **Rows Deleted –** Total number of records that were deleted from the audited table by the user or program specified in other report columns.
9. **Rows Updated –** Total number of records that were modified in the audited table by the user or program specified in other report columns

Audit Trail by Schema Report

This report returns consolidated detail information from the data audit trail for all audited tables in a particular database schema. To run the report, select the **Reports > Data Change Audit Trail > Audit Trail By Schema** menu. The **Select Audited Schema** dialog will appear. Choose the desired schema and click the **OK** button. The **Specify Report Filter** dialog will then appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the database audit trail tables.

The following columns are displayed on the report:

1. **Timestamp** – The date and time of the change.
2.  **Oracle, ASA, DB2: Table Name** – The full table name including schema name.
    **ASE, SQL Server: Table Name** – The full table name including database and schema names.
3. **Operation –** Type of database operation that caused the change; i.e., one of the following: INSERT, UPDATE or DELETE.

4. **Program Name –** The name of the program used to make changes.
5. **Terminal –** Network name of the terminal or user workstation where these changes were made.
6. **Oracle, ASA, DB2: Network User –** Network name of the user who made the change. If the database system uses trusted connections, this value will match the value in the **Database User** column.
7. **ASE, SQL Server: Login Name –** Login name of the user who made the change. If the database system uses trusted connections, this value will match the user's network name.
8. **Database User –** Name of the database user who made the change.

Audit Trail by Application Report

This report returns consolidated detail information from the data audit trail for all audited tables that were updated by a particular application. To run the report, select the **Reports > Data Change Audit Trail > Audit Trail By Application** menu. The **Enter Application Name** dialog will appear. Type in the desired application name or click the **Browse** button to paste the application name from an audited table, then click the **OK** button to continue. The **Specify Report Filter** dialog will then appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the database audit trail tables.

MySQL: This report is not available for MySQL database servers.

The following columns are displayed on the report:

1. **Timestamp** – The date and time of the change.
2. **Program Name –** The name of the program used to make changes.
3. **Oracle, ASA, DB2: Table Name** – The full table name including schema name.
   ASE, SQL Server: Table Name – The full table name including database and schema names.
4. **Operation –** Type of database operation that caused the change; i.e., one of the following: INSERT, UPDATE or DELETE.
5. **Terminal –** Network name of the terminal or user workstation where these changes were made.
6. **Oracle, ASA, DB2: Network User –** Network name of the user who made the change. If the database system uses trusted connections, this value will match the value in the **Database User** column.
7. **ASE, SQL Server: Login Name –** Login name of the user who made the change. If the database system uses trusted connections, this value will match the user's network name.
8. **Database User –** Name of the database user who made the change.

Enabled System Audits

Default System Audit Options Report

Oracle: This report is available only for Oracle databases.

This report returns Oracle default system audit options that will be used for newly created schema objects. To run the report, select the **Reports > System Audit Configuration > Default System Audit Options** menu. The following columns are displayed on the report:

1. **Operation** – The type of the operation to audit.
2. **Write to Audit Trail** – The value in this column indicates whether or not the operation is audited by default and is one of the following:
   Audit by Session Whenever Successful
   Audit by Session NOT Successful
   Audit by Access Whenever Successful
   Audit by Access Whenever NOT Successful
   Audit by Access Always
   Audit by Session Always
   None

Enabled Global Audit Options Report

To run the report, select the **Reports > System Audit Configuration > Enabled Global Audit Options** menu. The report output differs for different database systems.

ASE: This report returns enabled audit options for global system operations. The following columns are displayed on the report:

1. **Option –** The short name for the option
2. **Audit State –** The value in this column indicates whether the operations described by the option are audited or not. One of the following values can appear in this column:
   -- (this means none)
   Always
   On success
   On failure
3. **Description –** The description of the option

SQL Server: This report returns enabled audit options for global system operations. **T**his report is available only for SQL Server versions 2000 and up:

The following columns are displayed on the report:

1. **Option –** The short name for the option
2. **Audit State –** The value in this column indicates whether or not the operations described by the option are audited. One of the following values can appear in this column:
   -- (this means none)
   Always
   On success
   On failure
3. **Description –** The description of the option

DB2: This report returns enabled audit options for global system operations. **T**his report is available only for SQL Server versions 2000 and up:

The following columns are displayed on the report:

1. **Option –** The short name for the option
2. **Audit State –** The value in this column indicates whether or not the operations described by the option are audited. One of the following values can appear in this column:
   NOT AUDITED
   ALWAYS
   ON SUCCESS
   ON FAILURE
3. **Description –** The description of the option

MySQL: This report returns enabled audit options for global system operations. **T**his report is available only for SQL Server versions 2000 and up:

The following columns are displayed on the report:

1. **Option –** The group of related operations, presented as a comma separated list of operation names
2. **Audit State –** The value in this column indicates whether or not the operations described by the option are audited. One of the following values can appear in this column:
   NOT AUDITED
   ALWAYS
   ON SUCCESS
   ON FAILURE
3. **Description –** The description of the option

Enabled SQL Statement and Operations Audit Options Report

To run the report, select the **Reports > System Audit Configuration > Enabled SQL Statement and Operations Audit Options** menu. The report output differs for different database systems.

ASE, Oracle: This report is only applicable to Oracle and Sybase ASE database servers.

ASE: This report describes enabled audit options for SQL statements and database-level operations. The following columns are displayed on the report:

1. **Database –** The name of the database in which the option specified in the next column is used.
2. **Option –** The option name (same as the type of the audited operation).
3. **Audit State –** A value indicating whether or not the operations described by the option are audited. The value in this column can be one of the following:
   -- (this means none)
   Always
   On success
   On failure

Oracle: This report returns the current SQL commands being audited across the database. The following columns are displayed on the report:

1. **User Name –** The name of the database user for which the option specified in the next column is used. The value in this column can be one of the following:
   The name of an authenticated user
   ANY CLIENT (if access by a proxy on behalf of any client is being audited)
   NULL (if system wide auditing is being done)
2. **Audit Option –** Name of the system auditing option.
3. **Success –** The value in this column indicates the audit mode for successful operations. The value in this column can be one of the following:
   BY SESSION
   BY ACCESS
   NOT SET
4. **Failure –** The value in this column indicates the audit mode for unsuccessful operations. The value in this column can be one of the following:
   BY SESSION
   BY ACCESS
   NOT SET

Enabled Schema Object Audit Options Reports

To run the report, select the **Reports > System Audit Configuration > Enabled Schema Object Audit Options** menu. The report output differs for different database systems.

ASE, Oracle: This report is only applicable to Oracle and Sybase ASE database servers.

ASE: This report returns enabled audit options for schema objects. The following columns are displayed in the report:

1. **Database –** The name of the database containing the object.
2. **Schema –** Name of the database schema the audited object belongs to.
3. **Object Name –** The name of the object being audited.
4. **Object Type –** The type of the object.
5. **Option –** The option name that is the same as the type of the audited operation.
6. **Audit State –** The value in this column indicates whether the operations described by the option are audited or not. The value in this column can be one of the following:
   -- (this means none)
   Always
   On success
   On failure

Oracle: This report returns current database schema objects being audited by access type. The following columns are displayed on the report:

- **Object Name –** The full name of the object being audited, including schema name.
- **Select –** The audit mode for SELECT operations. The value in this column is one of the following:
  S/A - by Access whenever Successful
  -/A - by Access whenever NOT Successful
  S/S - by Session whenever Successful
  -/S - by Session whenever NOT Successful

- **Delete –** The audit mode for DELETE operations. The value in this column is one of those described for the **Select Audit mode**.
- **Update –** The audit mode for UPDATE operations. The value in this column is one of those described for the **Select Audit mode**.
- **Insert –** The audit mode for INSERT operations. The value in this column is one of those described for the **Select Audit mode**.
- **Execute –** The audit mode for EXECUTE operations. The value in this column is one of those described for the **Select Audit mode**.
- **Lock –** The audit mode for TABLE LOCK operations. The value in this column is one of those described for the **Select Audit mode**.
- **Alter –** The audit mode for ALTER operations. The value in this column is one of those described for the **Select Audit mode**.
- **Comment –** The audit mode for COMMENT ON <object> operations. The value in this column is one of those described for the **Select Audit mode**.
- **Index –** The audit mode for CREATE INDEX operations. The value in this column is one of those described for the **Select Audit mode**.
- **Reference –** The audit mode for operations involving referencing of the audited object. The value in this column is one of those described for the **Select Audit mode**.
- **Rename –** The audit mode for RENAME operations. The value in this column is one of those described for the **Select Audit mode**.
- **Audit –** The audit mode for AUDIT and NOAUDIT operations. The value in this column is one of those described for the **Select Audit mode**.
- **Grant –** The audit mode for GRANT and REVOKE operations. The value in this column is one of those described for the **Select Audit mode**.

The following additional columns appear on the report when connected to an Oracle 8i or later Database:

- **Create –** The audit mode for CREATE and DROP operations. The value in this column is one of those described for the **Select Audit mode**.
- **Read –** The audit mode for READ operations. The value in this column is one of those described for the **Select Audit mode**.
- **Write –** The audit mode for WRITE operations. The value in this column is one of those described for the **Select Audit mode**.

Enabled System Privilege Audit Options Report

To run the report, select the **Reports > System Audit Configuration > Enabled System Privilege Audit Options** menu.

 **Oracle:** This report is only applicable to Oracle database servers.

 **Oracle:** This report is available only for Oracle databases. The report describes the current system privilege being audited across the database and by user. The following columns are displayed on the report:

1. **User Name –** The name of the database user for which the option specified in the next column is used.

ANY CLIENT (if access by a proxy on behalf of any client is being audited)
NULL (if system wide auditing is being done)

2. **Privilege –** Name of the system privilege being audited.
3. **Success –** The value in this column indicates audit mode for successful privilege usage. The value in this column is one of the following:
   BY SESSION
   BY ACCESS
   NOT SET
4. **Failure –** The value in this column indicates audit mode for unsuccessful operations. The value in this column is one of the following:
   BY SESSION
   BY ACCESS
   NOT SET

Enabled Logon Audit Options Report

To run the report, select the **Reports > System Audit Configuration > Enabled Logon Audit Options** menu.

 **ASE:** This report is only applicable to Sybase ASE database servers.

 **ASE:** This report is available only for Adaptive Server Enterprise databases. The report describes enabled audit options for server logins. The following columns are displayed on the report:

1. **Login Name –** The server login name.
2. **Option –** The audit option name
3. **Audit State –** a Value indicating whether or not the operations described by the option are audited. The value in this column is one of the following:
   -- (this means none)
   Always
   On success
   On failure

System Audit and Security Reports

This group of reports provides you with powerful tools for performing forensic analysis of user and database activities. Many of the reports from other report groups are designed to find specific problems. The reports in this group can be used to further drill-down to determine exactly what user actions were taken and what was the state of the database before and after those actions were performed.

Logon/Logoff and Resource Usage Audit Report

To run the report, select the **Reports/System Audit Trail/Logon/Logoff and Resource Usage Audit** menu. The **Specify Report Filter** dialog will appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the system audit trail tables.

The report output differs for different database systems.

SQL Server, ASE: This report returns information for audited database connections and their duration. The following columns are displayed on the report:

1. **Login Name –** The server login name.
2. **OS User Name –** Network user name of the database user who attempted to establish a new database connection.

   ASE: This value is not available and is always reported as NULL.
3. **Login Time –** The date and time when the user attempted to establish a database connection.
4. **Logout Time –** The date and time when the user ended the database connection.

   SQL Server: A value of NULL is reported if the connection is still active or the server has been restarted without closing the connection.

   ASE: A value of NULL is reported if the connection was denied. If the user session terminated abnormally, this is the time when the connection was marked as dead.
5. **Terminal –** Name of the network terminal or user workstation from which the connection was made. This is the value reported by the Host connection parameter.
6. **Duration –** The duration of the database session.

Oracle: This report returns information for audited database connections, their duration and system resource usage. The following columns are displayed on the report:

1. **User Name –** Name of the database user who attempted to establish a database connection.
2. **OS User Name –** Network user name of the database user who attempted to establish a database connection.
3. **Logon Time –** The date and time when the user attempted to establish a database connection.
4. **Logoff Time –** The date and time when the user ended the database connection or NULL if the connection was denied. If the user session terminated abnormally, this is the time when the connection was marked as dead.
5. **Logical Reads –** The total number of logical reads for that session.
6. **Physical Reads –** The total number of physical reads for that session.
7. **Logical Writes –** The total number of logical writes for that session
8. **Dead Locks –** Deadlocks detected during that session, if any.

DB2: This report returns information for audited database connections and their duration. The following columns are displayed on the report:

1. **User Name –** Name of the database user who attempted to establish a new database connection.
2. **OS User Name –** Network user name of the database user who attempted to establish a new database connection.
3. **Terminal –** Name of the network terminal or user workstation from which the connection was made. For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
4. **Logon Time –** The date and time when the user attempted to establish a database

connection.

5. **Last Activity Time –** The date and time of the last session activity. This value should approximate session Logout Time.

MySQL: This report returns information for audited database connections and their duration. The following columns are displayed on the report:

1. **User Name –** Name of the database user who attempted to establish a database connection.
2. **Terminal –** Name of the network terminal or user workstation from which the connection was made as reported by MySQL client software.
3. **Logon Time –** The date and time when the user attempted to establish a database connection.
4. **Logout Time –** The date and time when the user ended the database connection.
5. **Duration –** The duration of the database session.

Object Access and Operations Audit Report

To run the report, select the **Reports > System Audit Trail > Object Access and Operations Audit** menu. The **Specify Report Filter** dialog will appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the system audit trail tables.

This report returns information for audited schema objects and operations performed on those objects. The following columns are displayed on the report:

Oracle:

1. **User Name –** The name of the database user.
2. **Schema –** Name of the database schema the audited object belongs to.
3. **Object Name –** The name of the object being audited.
4. **Action –** The type of the operation performed.
5. **Event Time –** The date and time when the operation was executed.

ASE:

1. **Login Name –** The server login name.
2. **Database –** The name of the database containing the object.
3. **Schema –** Name of the database schema the audited object belongs to.
4. **Object Name –** The name of the object being audited.
5. **Action –** The type of the operation performed.
6. **Event Time –** The date and time when the operation was executed.

SQL Server:

1. **Login Name –** The server login name.

2. **Database –** The name of the database containing the object.
3. **Schema –** Name of the database schema the audited object belongs to.
4. **Object Name –** The name of the object being audited.
5. **Action –** The type of the operation performed.
6. **Event Time –** The date and time when the operation was executed.
7. **Terminal –** Name of the network terminal or user workstation from which the connection was made. This is the value reported by the Host connection parameter.
8. **OS User Name –** Network user name of the database user who attempted to execute the operation.
9. **Program Name –** The name of the program from which the user attempted to execute the operation.

 **DB2:**

1. **User Name –** The name of the database user.
2. **Database –** The name of the database containing the object.
3. **Schema –** Name of the database schema the audited object belongs to.
4. **Object Name –** The name of the object being audited.
5. **Action –** The type of the operation performed.
6. **Reason –** The explanation of why access to the object or statement was granted or denied.
7. **Event Time –** The date and time when the operation was executed.
8. **Terminal –** Name of the network terminal or user workstation from which the connection was made. For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections ,the Terminal field displays [LOCAL]. For Network Pipe connections the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
9. **Program Name –** The name of the program from which the user attempted to execute the operation.

 **MySQL:**

1. **User Name –** The name of the database user.
2. **Schema –** Name of the database schema the audited object belongs to.
3. **Object Name –** The name of the object being audited.
4. **Action –** The type of the operation performed.
5. **Event Time –** The date and time when the operation was executed.
6. **Terminal –** Name of the network terminal or user workstation from which the connection was made.

Object Access Audit Summary Report

To run the report, select the **Reports > Object Access Audit Summary** menu. The **Specify Report Filter** dialog will appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the system audit trail tables.

This report returns summary information for audited schema objects and operations performed on those objects. The following columns are displayed on the report:

 **Oracle:**

1. **User Name –** The name of the database user.
2. **OS User Name –** Network user name of the database user who attempted to establish a new database connection.
3. **Schema –** Name of the database schema the audited object belongs to.
4. **Object Name –** The name of the object being audited.
5. **Total Success –** The total number of successful object accesses. This number includes a count of all **audited operations** that refer to the object directly. Indirect references such as access from within stored procedures, triggers, referential constrains, and so on, are not counted.
6. **Total Failure -** The total number of failed object accesses. This number includes a count of all **audited operations** that refer to the object directly. Indirect references such as access from within stored procedures, triggers, referential constrains, and so on, are not counted.

 **ASE:**

1. **Login Name –** The server login name.
2. **Database –** The name of the database containing the object.
3. **Schema –** Name of the database schema the audited object belongs to.
4. **Object Name –** The name of the object being audited.
5. **Total Success –** The total number of successful object accesses. This number includes a count of all **audited operations** that refer to the object including direct references and indirect references such as access from within stored procedures, triggers, referential constrains, etc. Every access is counted. If an object is accessed multiple times from the same procedure, it is counted multiple times. However if it is accessed using a cursor, it is counted only once (when the cursor is opened).
6. **Total Failure -** The total number of failed object accesses. This number includes a count of all **audited operations** that refer to the object including direct references and indirect references such as access from within stored procedures, triggers, referential constrains, etc…. Every access is counted. If an object is accessed multiple times from the same procedure, it is counted multiple times. However if it is accessed using a cursor, it is counted only once (when the cursor is opened).

 **SQL Server:**

1. **Login Name –** The server login name.
2. **Database –** The name of the database containing the object.
3. **Schema –** Name of the database schema the audited object belongs to.
4. **Object Name –** The name of the object being audited.

5. **Total–** The total number of object accesses. This number includes a count of all **audited operations** that refer to the object, including direct references and indirect references such as access from within stored procedures, triggers, referential constrains, etc. Every access is counted. If an object is accessed multiple times from the same procedure, it is counted multiple times. However if it is accessed using a cursor, it is counted only once (when the cursor is opened).
6. **Total Top-Level –** The total number of direct object accesses. This number includes a count of all **audited operations** that refer to the object directly. Nested-level access from procedures, triggers, and so on, is not counted.

 **DB2:**

1. **User Name –** The name of the database user.
2. **Database –** The name of the database containing the object.
3. **Schema –** Name of the database schema the audited object belongs to.
4. **Object Name –** The name of the object being audited.
5. **Total–** The total number of object accesses. This number includes a count of all **audited operations** that refer to the object directly. Indirect references such as access from within stored procedures, triggers, referential constrains, and so on, are not counted.

 **MySQL:**

1. **User Name –** The name of the database user.
2. **Schema –** Name of the database schema the audited object belongs to.
3. **Object Name –** The name of the object being audited.
4. **Total Success–** The total number of successful object accesses. This number includes a count of all **audited operations** that refer to the object directly as well as any indirect references such as access from within stored procedures and triggers.
5. **Total Failure –** The total number of failed object accesses. Failed access could be a result of insufficient user privileges or other problems. This number includes a count of all **audited operations** that refer to the object directly as well as any indirect references such as access from within stored procedures and triggers.

Operations Audit Detail

To run the report, select the **Reports > Operations Audit Detail** menu. The **Specify Report Filter** dialog will appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the system audit trail tables.

This report returns detailed audit trail information for audited schema objects and operations performed on those objects. The following columns are displayed on the report:

 **Oracle:**

1. **User Name –** The name of the database user.
2. **OS User Name –** Network user name of the database user.
3. **Host –** Name of the network computer or application server from which the connection to the

database was established.

4. **Terminal –** Name of the network terminal or user workstation on which the database operation was initiated.
5. **Event Time –** The date and time when the operation was executed.
6. **Action –** The type of operation performed.
7. **Schema –** Name of the database schema the audited object belongs to.
8. **Object Name –** The name of the object being audited.
9. **New Name –** The new object name for rename operations; blank for all other operation types.
10. **Object Privilege –** The object-level privilege updated, granted or revoked by the user as a result of the operation.
11. **System Privilege –** The system-level privilege updated, granted or revoked by the user as a result of the operation.
12. **Return Code –** The return code of the operation; zero (0) if the operation was successful. See your Oracle documentation for description a description of any nonzero errors returned.

SQL Server:

1. **OS User Name –** Network user name of the database user.
2. **Login Name –** The server login name.
3. **Host -** Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
4. **Program Name -** The name of the program that was running when the user attempted to execute the operation.
5. **Event Time –** The date and time when the operation was executed.
6. **Action –** The type of the operation performed.
7. **Database –** The name of the database containing the object.
8. **Schema –** Name of the database schema the audited object belongs to.
9. **Object Name –** The name of the object being audited.

DB2:

1. **OS User Name –** Network user name of the database user.
2. **User Name –** The name of the database user.
3. **Program Name -** The name of the program that was running when the user attempted to execute the operation.
4. **Event Time –** The date and time when the operation was executed.
5. **Action –** The type of the operation performed.
6. **Database –** The name of the database containing the object.
7. **Schema –** Name of the database schema the audited object belongs to.
8. **Object Name –** The name of the object being audited.
9. **Grantee–** The object-level or system-level privilege updated, granted or revoked by the user as a result of the operation.

10. **Return Code –** The return code of the operation; zero (0) if the operation was successful or a non-zero number indicating DB2 processing error. See your Oracle documentation for description a description of any nonzero errors returned.

MySQL:

1. **User Name –** The name of the database user.
2. **Terminal –** Name of the network terminal or user workstation on which the database operation was initiated.
3. **Event Time –** The date and time when the operation was executed.
4. **Action –** The type of the operation performed.
5. **Schema –** Name of the database schema the audited object belongs to.
6. **Object Name –** The name of the object being audited.

Operations Audit Summary

To run the report, select the **Reports > Operations Audit Summary** menu. The **Specify Report Filter** dialog will appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the system audit trail tables.

This report returns summary audit trail information for audited database operations. The following columns are displayed on the report:

Oracle:

1. **User Name –** The name of the database user.
2. **Action –** The type of the operation executed.
3. **Total Success–** The total number of successful operation executions. This number includes a count of **audited operations** only. Unaudited operations do not appear on the report.
4. **Total Failure–** The total number of failed operation executions. This number includes a count of **audited operations** only. Unaudited operations do not appear on the report.

ASE:

1. **Login Name –** The server login name.
2. **Database –** The name of the database where the operation was executed.
3. **Action –** The type of the operation executed.
4. **Total Success–** The total number of successful operation executions. This number includes a count of **audited operations** only. Unaudited operations do not appear on the report.
5. **Total Failure –** The total number of failed operation executions. This number includes a count of **audited operations** only. Unaudited operations do not appear on the report.

SQL Server:

1. **Login Name –** The server login name.
2. **Database –** The name of the database where the operation was executed.
3. **Action –** The type of the operation executed.
4. **Total–** The total number of operation executions. This number includes count for all **audited operations** executed directly by the user program or indirectly as nested operations executed within called stored procedures, triggers and so on. Unaudited operations do not appear on the report.
5. **Total Top-Level –** The total number of operation executions. This number includes a count of all **audited operations** executed directly by the user program. Indirect nested operations executed within called stored procedures, triggers and so on, are not included into this count. Unaudited operations do not appear on the report.

DB2:

1. **User Name –** The name of the database user.
2. **Database –** The name of the database where the operation was executed.
3. **Action –** The type of the operation executed.
4. **Total –** The total number of operation executions. This number includes a count of **audited operations** only. Unaudited operations do not appear on the report.

MySQL:

1. **User Name –** The name of the database user.
2. **Terminal –** Name of the network terminal or user workstation on which the database operation was initiated.
3. **Action –** The type of the operation executed.
4. **Total Success –** The total number of successful operation executions. This number includes a count for all **audited operations** only. Not audited operations do not appear on the report.
5. **Total Failure –** The total number of failed operation executions. This number includes count of all **audited operations** only. Unaudited operations do not appear on the report.

User Activity (Failed Logons) Report

To run the report, select the **Reports > System Audit Trail > User Activity (Failed Logons)** menu. This report analyzes data from the system audit trail and returns names of users/logins who were denied database access for any reason.

While the report may indicate possible hacking attempts, it could also be a result of harmless attempts to enter an incorrect password. The Auditor should use personal judgment to decide which of the above situations occurred.

SQL Server: This report is available only for versions 2000 and up:

The report output differs for different database systems. The following columns are displayed on the report:

ASE:

1. **Login Time –** The date and time when the attempt to establish a new database connection failed.
2. **Login Name –** Name of the server login who attempted to establish a new database connection.

SQL Server:

1. **Login Time –** The date and time when the attempt to establish a new database connection failed.
2. **OS User Name –** Network user name of the database user who attempted to establish a new database connection.
3. **Login Name –** Name of the server login who attempted to establish a new database connection.

Oracle, DB2:

1. **Login Time –** The date and time when the attempt to establish a new database connection failed.
2. **User Name –** Name of the database user who attempted to establish a new database connection.
3. **OS User Name –** Network user name of the database user who attempted to establish a new database connection.
4. **Terminal –** Name of the network terminal or user workstation from which the attempt to make a new connection failed.

   For DB2: for TCP/IP connections, the Terminal field displays the IP address of the user's terminal; For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types it displays the application ID, including connection handles, as reported by DB2 audit.
5. **Connection ID – A u**nique connection identifier describing the connection method and handles.

MySQL:

1. **Login Time –** The date and time when the attempt to establish a new database connection failed.
2. **User Name –** Name of the database user who attempted to establish a new database connection.
3. **Terminal –** Name of the network terminal or user workstation from which the attempt to make a new connection failed.

User Activity (Last Logon Time) Report

To run the report, select the **Reports > System Audit Trail > User Activity (Last Logon Time)** menu. This report analyzes data from the system audit trail and returns names of users/logins and time when they last successfully connected to the database.

SQL Server: This report is available only for versions 2000 and up:

The report output differs for different database systems. The following columns are displayed on the report:

SQL Server, ASE:

1. **Login Name –** Name of the server login who established at least one successful connection to the database.
2. **Last Login Time –** Date and time of the last successful connection.
3. **Days –** Number of days elapsed since the last successful connection.

Oracle, DB2:

1. **User Name –** Name of the database user who established at least one successful connection to the database.
2. **OS User Name –** Network user name of the database user who at least once established a successful connection to the database.
3. **Last Login Time –** Date and time of the last successful connection.
4. **Days –** Number of days elapsed since the last successful connection.

MySQL:

1. **User Name –** Name of the database user who established at least one successful connection to the database.
2. **Last Login Time –** Date and time of the last successful connection.
3. **Days –** Number of days elapsed since the last successful connection.

User Activity (Denied Access to Objects) Report

To run the report, select the **Reports > System Audit Trail > User Activity (Denied Access to Objects)** menu. This report analyzes data from the system audit trail and lists possible security violations. The columns listed below are displayed on the report:

While this report may indicate possible hacking attempts, it could also be a result of harmless attempts to access an object or set of objects that were previously accessible for that particular user. The Auditor should use personal judgment to decide which of the above situations occurred. For example, if a user moved from one department to another, he/she may now have some privileges removed due to the new responsibilities, but he/she may still be attempting to access previously available data.

SQL Server: This report is available only for versions 2000 and up:

The report output differs for different database systems. The following columns are displayed on the report:

Oracle:

1. **Event Time –** The date and time when the violation occurred.
2. **Action** – The type of action attempted by the user.

3. **Object Name –** The full object name including schema name.
4. **User Name –** Name of the database user who attempted to perform the specified action.
5. **OS User Name –** Network user name of the database user who attempted to perform the specified action.
6. **Terminal –** Name of the network terminal or user workstation on which the failed attempt was made.

 **ASE:**

1. **Event Time –** The date and time when the violation occurred.
2. **Action** – The type of action attempted by the user.
3. **Database –** The name of the database containing the object.
4. **Schema –** Name of the database schema the audited object belongs to.
5. **Object Name –** The name of the object being audited.
6. **Login Name –** Login name of the database user who attempted to perform the specified action.

 **SQL Server:**

1. **Event Time –** The date and time when the violation occurred.
2. **SPID** – The system process ID (session ID).
3. **Login Name –** Login name of the database user who attempted to perform the specified action.
4. **OS User Name –** Network user name of the database user who attempted to perform the specified action.
5. **Terminal –** Name of the network terminal or user workstation from which the connection was made. This is the value reported by the Host connection parameter.
6. **Program Name –** The name of the program that was running when the error occurred.
7. **Database –** The name of the database containing the object.
8. **Object Type** – The type of the object being audited.
9. **Schema –** Name of the database schema the audited object belongs to.
10. **Object Name –** The name of the object being audited.
11. **SQL Text –** Text of the SQL query that attempted to access the audited object.

 **DB2:**

1. **Event Time –** The date and time when the violation occurred.
2. **Connection ID – A u**nique connection identifier describing the connection method and handles.
3. **User Name –** Name of the database user who attempted to perform the specified action.
4. **OS User Name –** Network user name of the database user who attempted to perform the specified action.

5. **Terminal –** Name of the network terminal or user workstation on which the failed attempt was made. For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
6. **Program Name –** The name of the program that was running when the error occurred.
7. **Database –** The name of the database containing the object.
8. **Object Type** – The type of the object being audited.
9. **Schema –** Name of the database schema the audited object belongs to.
10. **Object Name –** The name of the object being audited.
11. **Action** – The type of action attempted by the user.

MySQL:

1. **Event Time –** The date and time when the violation occurred.
2. **SPID –** The connect threat ID (session ID).
3. **User Name –** Name of the database user who attempted to perform the specified action.
4. **Terminal –** Name of the network terminal or user workstation on which the failed attempt was made.
5. **Object Type** – The type of the object being audited.
6. **Schema –** Name of the database schema the audited object belongs to.
7. **Object Name –** The name of the object being audited.
8. **Action** – The type of action attempted by the user.
9. **SQL Text –** Text of the SQL query that attempted to access the audited object.

User Activity (Sys Admins) Report

To run the report, select the **Reports > System Audit Trail > User Activity (Sys Admins)** menu.

Notes: This report is specific to DB2 and Oracle databases; it cannot be used with other database systems.

Oracle:

When a user with SYSDBA or SYSOPER privileges connects to the database, the action is expected to be for administrative reasons such as database shutdown, startup, configuration changes, etc. These actions are assumed not to require auditing so the Oracle auditing mechanism does not pick them up.

Starting with Oracle release 9iR2, the database provides the full audit ability of the SYS account. In all versions prior to 9iR2, the only auditing option available was the small audit file that Oracle creates by default in the directory $ORACLE_HOME/RDBMS/audit (in a Unix environment) or as

a brief audit record written to the Windows NT application Event Log in Windows environments. This file is created every time a user attempts to connect internally through server manager (svrmgrl) or the now current method of connecting as SYSDBA or SYSOPER. A separate file is written for every connection making it difficult to analyze the audit trail data and monitor security breaches.

To close the gap, DB Audit provides methods for monitoring Oracle audit files and loading them into a database table which can be then queried and used for reporting. For information on how to install and configure system audit files loading and processing procedures, see CHAPTER 3, Configuring Advanced Options for Oracle.

This report analyzes data from the system audit table described above and lists possible security violations. The following columns are displayed on the report:

1. **OS User Name –** Network user name of the database user who attempted to perform the specified action.
2. **User Name –** Name of the database user who attempted to perform the specified action.
3. **Program Name –** The name of the program that was running when the error occurred.
4. **Event Time –** The date and time when the event occurred.
5. **Terminal –** Name of the network terminal or user workstation on which the failed attempt was made.
6. **Action / Query** – The type of action attempted by the user and, for SQL query actions, the first 255 characters of the query text,
7. **Privileged Used –** The type of database or operation system privilege used to perform the audited operation; for example, OPER or DBA.
8. **Status** – The return code of the attempted action. If the action is successful, the return code is reported as SUCCESS; otherwise, it is reported as FAILED followed by the Oracle error number.

 **DB2:**

This report displays audit records for operations requiring SYSADM, SYSMAINT, or SYSCTRL authority. The following columns are displayed on the report:

1. **OS User Name – The n**etwork user name of the database user who attempted to perform the specified action.
2. **User Name – The n**ame of the database user who attempted to perform the specified action.
3. **Program Name –** The name of the program that was running when the error occurred.
4. **Event Time –** The date and time when the event occurred.
5. **Database –** The name of the database where the event occurred; blank if the event occurred at the instance level.
6. **Terminal –** Name of the network terminal or user workstation on which the failed attempt was made. For TCP/IP connections, the Terminal field displays the IP address of the user's terminal; For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections. The Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
7. **Action** – The type of action attempted by the user.
8. **Return Code** – The return code of the attempted action. If the action is successful, the return code is reported as SUCCESS; otherwise, the actual DB2 error number is reported.

Possible action values:

START DB2
STOP DB2
CREATE DATABASE
ALTER DATABASE
DROP DATABASE
UPDATE DBM CFG
UPDATE DB CFG
CREATE TABLESPACE
DROP TABLESPACE
ALTER TABLESPACE
RENAME TABLESPACE
CREATE NODEGROUP
DROP NODEGROUP
ALTER NODEGROUP
CREATE BUFFERPOOL
DROP BUFFERPOOL
ALTER BUFFERPOOL
CREATE EVENT MONITOR
DROP EVENT MONITOR
ENABLE MULTIPAGE
MIGRATE DB DIR
DB2TRC
DB2SET
ACTIVATE DB
ADD NODE
BACKUP DB
CATALOG NODE
CATALOG DB
CATALOG DCS DB
CHANGE DB COMMENT
DEACTIVATE DB
DROP NODE VERIFY
FORCE APPLICATION
GET SNAPSHOT
LIST DRDA INDOUBT TRANSACTIONS
MIGRATE DB
RESET ADMIN CFG
RESET DB CFG
RESET DBM CFG
RESET MONITOR
RESTORE DB
ROLLFORWARD DB
SET RUNTIME DEGREE
SET TABLESPACE CONTAINERS
UNCATALOG DB
UNCATALOG DCS DB
UNCATALOG NODE
UPDATE ADMIN CFG
UPDATE MON SWITCHES
LOAD TABLE
DB2AUDIT
SET APPL PRIORITY
CREATE DB AT NODE
KILLDBM
MIGRATE SYSTEM DIRECTORY
DB2REMOT
DB2AUD
MERGE DBM CONFIG FILE

UPDATE CLI CONFIGURATION
OPEN TABLESPACE QUERY
SINGLE TABLESPACE QUERY
CLOSE TABLESPACE QUERY
FETCH TABLESPACE
OPEN CONTAINER QUERY
FETCH CONTAINER QUERY
CLOSE CONTAINER QUERY
GET TABLESPACE STATISTICS
DESCRIBE DATABASE
ESTIMATE SNAPSHOT SIZE
READ ASYNC LOG RECORD
PRUNE RECOVERY HISTORY
UPDATE RECOVERY HISTORY
QUIESCE TABLESPACE
UNLOAD TABLE
UPDATE DATABASE VERSION
CREATE INSTANCE
DELETE INSTANCE
SET EVENT MONITOR
GRANT DBADM
REVOKE DBADM
GRANT DB AUTHORITIES
REVOKE DB AUTHORITIES
REDIST NODEGROUP

Database Errors Report

To run the report, select the **Reports > System Audit Trail > Database Errors** menu.

Oracle: Database error capturing and logging procedures must be installed using Advanced Audit Options menu. For information on how to install and configure database error capturing and logging procedures, see Advanced Options topic in CHAPTER 3.

SQL Server: Audit Errors and Audit Exceptions types of operations must be enabled in the system audit options in order to enable DB Audit to collect error events. This report is available only for Microsoft SQL Server 2000 (and up).

DB2: Only errors occurred in audited operations appear on this report. Errors occurred in unaudited operations are not recoded and do not appear on the report.

MySQL: Only errors occurred in audited operations appear on this report. Errors occurred in unaudited operations are not recoded and do not appear on the report.

The following columns are displayed on the report:

Oracle:

1. **Error # –** The error number reported by Oracle.
2. **Time** – The date and time when the error occurred.
3. **Error Message –** The error message reported by Oracle.

4. **SID –** The Session ID of the Oracle user who received the error.
5. **User Name** – Name of the database user who received the error.
6. **OS User Name** – Network user name of the database user who received the error.
7. **Program Name –** The name of the program that was running when the error occurred.
8. **Terminal –** Network name of the terminal or user workstation on which the user program was run.
9. **Operation –** Type of the operation that led to the error, such as CREATE, DROP, etc.
10. **Object Type –** The type of object that user attempted to access when the error occurred.
11. **Schema –** Name of the user schema containing the error object.
12. **Object Name –** Name of the error object the user attempted to access when the error occurred.
13. **Oracle 9.0 and later: SQL Query –** Text of the SQL query (first 4000 characters) that caused the error.

SQL Server:

1. **Error # –** The error number reported by SQL Server.
2. **Time** – The date and time when the error occurred.
3. **Error Message –** The error message reported by SQL Server.
4. **SPID –** The system process ID (Session ID) of the SQL Server user who received the error.
5. **Login Name** – Login name of the user who received the error.
6. **OS User Name** – Network user name of the database user who received the error.
7. **Program Name –** The name of the program that was running when the error occurred.
8. **Terminal –** Network name of the terminal or user workstation on which the user program was run.
9. **Database –** The name of the database containing the object that the user attempted to access when the error occurred.
10. **Object Type –** The type of object user attempted to access when the error occurred.
11. **Schema –** Name of the user schema containing the error object.
12. **Object Name –** Name of the error object the user attempted to access when the error occurred.
13. **Message Data –** Additional data related to the error message.

DB2:

1. **OS User Name – The n**etwork user name of the database user.
2. **User Name –** The name of the database user.
3. **Program Name -** The name of the program that was running when the user attempted to execute the operation.
4. **Timestamp –** The date and time when the operation was executed.
5. **Action –** The type of operation performed.

6. **Database –** The name of the database containing the object.
7. **Schema –** Name of the database schema the audited object belongs to.
8. **Object Name –** The name of the object being audited.
9. **Return Code –** The DB2 error number. See your DB2 documentation for description of the error.

MySQL:

1. **Error # –** The error number reported by MySQL server.
2. **Time** – The date and time when the error occurred.
3. **Error Message –** The error message associated with the error
4. **SPID –** The system thread ID (Session ID) of the MySQL user who received the error.
5. **User Name** – Name of the database user who received the error.
6. **Terminal –** Network name of the terminal or user workstation where the user program was run.
7. **Object Type –** The type of object user attempted to access when the error occurred.
8. **Schema –** The name of the user schema containing the error object.
9. **Object Name –** The name of the error object user attempted to access when the error occurred.
10. **SQL Command –** The text of SQL query in which the error occurred (first 32000 bytes).

Text of SQL Queries Report

To run the report, select the **Reports > System Audit Configuration > Text of SQL Queries** menu. The **Specify Report Filter** dialog will appear. If you wish, you may enter the optional report filter to limit the amount of data retrieved from the database audit trail tables.

The report returns the text of SQL queries executed on the server. The text is recorded in the system audit tables regardless of whether those queries succeeded or failed.

Oracle: The following columns are displayed on the report:

1. **OS User Name** – Network user name of the database user who attempted to execute the SQL query.
2. **User Name** – Name of the database user who attempted to execute the SQL query.
3. **Terminal –** Network name of the terminal or user workstation on which the program was running.
4. **Time –** The date and time when the SQL commands were executed.
5. **Return Code –** The return code of the operation; zero (0) if the operation was successful. See your Oracle documentation for a description of non-zero error codes returned.
6. **SQL Text –** The text of the SQL query being executed

SQL Server: This report is not available with Microsoft SQL Server 7 and earlier versions. The following columns are displayed on the report:

1. **Login Name –** Login name of the database user who attempted to execute the SQL query.
2. **SPID –** Database system process ID that is used to identify the user session.
3. **Program Name –** The name of the program from which the query was sent to the database server.
4. **Database –** The session database from which the SQL was executed.
5. **Time –** The date and time when the SQL was executed.
6. **SQL Text –** The text of the SQL query being executed (first 2000 bytes).

ASE: The following columns are displayed on the report:

1. **Login Name –** Login name of the database user who attempted to execute the SQL query.
2. **SPID –** Database system process ID used to identify the user session.
3. **Database –** The session database from which the SQL was executed.
4. **Time –** The date and time when the SQL was executed.
5. **SQL Text –** The text of the SQL query being executed. The full query text may occupy more than one physical line and more than one report row (logical line).

   Each logical line is prefixed with a four-digit line number.

DB2: Only successfully executed queries are available in the audit trail for this type of report, so only successful queries can be reported.

The following columns are displayed on the report:

1. **User Name –** Login name of the database user who executed the SQL query.
2. **Connection ID – A u**nique connection identifier that describes the connection method and handles.
3. **Program Name –** The name of the program from which the query was sent to the database server.
4. **Database –** The session database where the SQL was executed.
5. **Time –** The date and time when the SQL was executed.
6. **SQL Text –** The text of the SQL query being executed (first 2000 bytes). The query text may occupy more than one physical line and more than one report row (logical line).

MySQL: The following columns are displayed on the report:

1. **User Name** – Name of the database user who executed the SQL query
2. **SPID –** Database system process ID used to identify the user session.
3. **Schema Name –** The name of the current database schema from which the query was executed.
4. **Time –** The date and time when the SQL was executed.
5. **SQL Text –** The actual text of the SQL query being executed (first 32000 bytes).

Compliance Reports

Compliance reports enable you to identify and monitor overall security compliance and control processes for critical areas of your database operations. These reports allow you to evaluate your database activities and settings and determine whether they are compliant with your organization security policies.

Recently Created, Deleted and Modified Users and Logins

To run the report, select the **Reports > Compliance Reports > Recently Created, Deleted and Modified Users and Logins** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays all attempts to create, delete or otherwise change login and user settings executed on the server during the specified reporting period, regardless of whether or note those changes succeeded or failed.

Note: The report output differs for different database systems. Specifics for each database system are described below.

SQL Server: This report is available only for version 2000 and up.

Requirements:

- DB Audit system auditing must be installed
- Auditing of both successful and unsuccessful login attempts must be enabled
- Add/Drop Login auditing must be enabled
- Add Database User auditing must be enabled
- Grant/Deny/Revoke Login auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **Login Name –** The login name of the user who attempted to execute the operation.
3. **Database –** The name of the database where operation was attempted (not applicable to operations on logins).
4. **Target Name –** The name of the affected login or user account.
5. **Action –** The type of the operation attempted.
6. **Terminal -** Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
7. **OS User Name –** Network user name of the database user.
8. **Program -** The name of the program that was running when the user attempted to execute the operation.
9. **SQL Command –** The actual SQL command that was attempted to execute on the server.
10. **Success –** The Yes or No status of the operation.

 **Oracle:** This report is available for version 7.3 and up.

Requirements:

- System auditing must be enabled
- CREATE USER, ALTER USER, DROP USER auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **User Name –** The name of the database user who attempted to execute the operation.
3. **Action –** The type of the operation attempted.
4. **Target Name –** The name of the affected user account.
5. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.
6. **OS User Name –** Network user name of the database user.
7. **Privileges Used –** For a successful operation, this column indicates which privileges allowed the user to perform this operation.
8. **Success –** The Yes/No status of the operation.

 **ASE:** This report is available only for version 11.5 and up.

Requirements:

- System auditing must be installed
- LOGINS auditing must be enabled

- SQL Text auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **Login Name –** The login name of the user who attempted to execute the operation.
3. **Action –** The type of operation attempted.
4. **Additional Info –** The text of the SQL command used to modify logins and/or privileges used.

 **DB2:** This report is not available for DB2; DB2 does not maintain internal database users. DB2 relies on the Operation System security functions for creating and maintaining users.

 **MySQL:** This report is available only for version 5.0.2 and up.

Requirements:

- DB Audit system auditing must be installed
- Auditing of both successful and unsuccessful logins must be enabled
- Create/Drop/Rename User auditing must be enabled
- Auditing of INSERT, DELETE, UPDATE operations for system tables in a MYSQL database schema is highly recommended
- Data must be available in the audit trail for the specified reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **User Name –** The name of the database user who attempted to execute the operation.
3. **Target Name –** The name of the affected user account.
4. **Action –** The type of operation attempted.
5. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.
6. **SQL Command –** The SQL command that was attempted to execute on the server.
7. **Success –** The Yes/No status of the operation.

Recently Granted and Revoked Privileges

To run the report, select the **Reports > Compliance Reports > Recently Granted and Revoked Privileges** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default, the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays all attempts to create, delete or otherwise change login and user privileges during the specified reporting period, regardless of whether those changes succeeded or failed.

Note: The report output differs for different database systems. Specifics for each database system are described below.

SQL Server: This report is available only for versions 2000 and up.

Requirements:

- DB Audit system auditing must be installed
- Grant/Deny/Revoke Privilege auditing must be enabled
- Grant/Deny/Revoke Object Access auditing must be enabled
- Grant/Deny/Revoke Login auditing must be enabled
- Add Database User auditing must be enabled
- Add/Drop Role Member auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **Login Name –** The login name of the user who attempted to execute the operation.
3. **Database –** The name of the database where operation was attempted (not applicable to operations on logins).
4. **Action –** The type of operation attempted.
5. **Grantee –** The name of the login, user or role whose privileges were affected.
6. **Terminal -** Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
7. **OS User Name –** Network user name of the database user.
8. **Program -** The name of the program from which the user attempted to execute the operation.
9. **SQL Command –** The actual SQL command that was attempted to execute on the server.
10. **Success –** The Yes/No status of the operation.

Oracle: This report is available for version 7.3 and up.

Requirements:

- System auditing must be enabled
- All GRANT and REVOKE statement-level operations must be enabled. Note that enabling object-level only GRANT/REVOKE auditing will not allow you to have the complete report reflecting all of the privilege changes in the database.
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.

2. **User Name –** The name of the database user who attempted to execute the operation.
3. **Action –** The type of the operation attempted.
4. **Grantee –** The name of the user or role whose privileges were affected.
5. **Target Name –** Name of the database object affected. Applicable to object-level privileges only.
6. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.
7. **OS User Name –** Network user name of the database user.
8. **Privileges Used –** For a successful operation, this column lists the privileges that allowed the user to perform this operation.
9. **Success –** The Yes/No status of the operation.

 **ASE**: This report is available only for version 11.5 and up.

Requirements:

- System auditing must be installed
- GRANT and REVOKE auditing must be enabled for each monitored database
- SECURITY auditing must be enabled
- SQL Text auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **Login Name –** The login name of the user who attempted to execute the operation.
3. **Action –** The type of the operation attempted.
4. **Database –** The name of the database where operation was attempted (not applicable to operations on logins).
5. **Schema –** Schema name of the database object affected. Applicable to object-level privileges only.
6. **Object Name -** Name of the database object affected. Applicable to object-level privileges only.
7. **Additional Info –** The text of the actual SQL command used to modify privileges and/or privileges used.

 **DB2:** This report is available only for DB2 UDB for Linux, Unix and Windows version 7.1 and up.

Requirements:

- DB Audit system auditing must be installed

- Security Changes auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **User Name –** The name of the database user who attempted to execute the operation.
3. **Grantee –** The name of the user, user group or role whose privileges were affected.
4. **Action –** The type of operation attempted.
5. **Database –** The name of the database where operation was attempted (not applicable to operations on logins).
6. **Schema –** Schema name of the database object affected. Applicable to object-level privileges only.
7. **Object Name -** Name of the database object affected. Applicable to object-level privileges only.
8. **Privilege –** Description of the privilege being granted or revoked.
9. **Terminal -** Name of the network terminal or user workstation from which the database connection was made. For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
10. **Program -** The name of the program from which the user attempted to execute the operation.
11. **Success –** The Yes/No status of the operation.

 **MySQL:** This report is available only for MySQL version 5.0.2 and up.

Requirements:

- DB Audit system auditing must be installed
- GRANT and REVOKE auditing must be enabled
- Auditing of INSERT, DELETE, UPDATE operations for system tables in MYSQL database schema is highly recommended
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Event Time –** The date and time when the operation was executed.
2. **User Name –** The name of the database user who attempted to execute the operation.
3. **Target Name –** The name of the user, or objects whose privileges were affected.
4. **Action –** The type of the operation attempted.
5. **Terminal –** Name of the network terminal or user workstation from which the database connection was made.
6. **SQL Command –** The actual SQL command that attempted to execute on the server.
7. **Success –** The Yes/No status of the operation.

Inactive Users with Active Accounts

To run the report, select the **Reports > Compliance Reports > Inactive Users with Active Accounts** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to users who did not login to the database during the previous two months (60 or 61 days depending on the calendar month).

The report lists all users (or logins) who have not used the database for a long time and whose connect privilege and other database privileges should be revoked.

 **Note:** The report output differs for different database systems. Specifics for each database system are described below.

 **SQL Server:** This report is available only for versions 2000 and up.

Requirements:

- DB Audit system auditing must be installed
- Both Login (failed) and Login (successful) event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Name –** The login name.
2. **Last Login Time –** The date and time of the last login.
3. **Days Inactive –** The number of days since the last login.

 **Oracle**: This report is available for versions 8.0 and up.

Requirements:

- System auditing must be enabled
- Connect or Session auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **Last Login Time –** The date and time of the last login.
3. **Days Inactive –** The number of days since the last login.

 **ASE**: This report is available for versions 11.5 and up.

Requirements:

- System auditing must be enabled
- LOGINS auditing must be enabled

- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Name –** The login name.
2. **Last Login Time –** The date and time of the last login.
3. **Days Inactive –** The number of days since the last login.

DB2: This report is available only for DB2 UDB for Linux, Unix and Windows versions 7.1 and up.

Requirements:

- DB Audit system auditing must be installed
- User Authentication auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **Last Login Time –** The date and time of the last login.
3. **Days Inactive –** The number of days since the last login.
4. **System Privileges –** Indicates if the user has been granted system-wide privileges.
5. **Database Privileges –** Indicates if the user has been granted database-wide privileges.
6. **Schema Privileges –** Indicates if the user has been granted CREATE SCHEMA privileges as well as access to existing tables.
7. **Table Privileges –** Indicates if the user has been granted CREATE TABLE privileges as well as access to existing tables.

MySQL: This report is available for versions 5.0.2 and up.

Requirements:

- System auditing must be enabled
- Both Login (failed) and Login (successful) event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **Last Login Time –** The date and time of the last login.
3. **Days Inactive –** The number of days since the last login.

Users with Expired Passwords

To run the report, select the **Reports > Compliance Reports > Users with Expired Passwords** menu.

The report lists all database users (or logins) having expired passwords.

 **Important Notes:**

- The scope of this report is limited to internal database users and logins only. It does not cover externally authenticated users connecting to the database using various types of "integrated security" and "enterprise login" methods.
- The report output differs for different database systems. Specifics for each database system are described below.

 **SQL Server:** This report is not available for SQL Server because the database server does not handle password expiration internally. Database logins based on SQL Server internal accounts never expire, while SQL Server logins based on external domain user accounts rely on the operating system for password validation and management.

 **Oracle**: This report is available for versions 7.3 and up.

Requirement: None

Notes: In Oracle, each user account is associated with a resource profile. The profile settings control password expiration and validation policies for associated user accounts. Multiple user accounts can be associated with a single profile.

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **User External Name –** The external user name, usually full user name or operating system user name, as specified in the user account settings.
3. **Password Expiration –** The password expiration option for the user account. If this option is set, it indicates the account has expired according to account settings; otherwise the "account expired" state has been set manually by a database administrator or other person with sufficient account management authority.
4. **Password Complexity –** The password complexity option for the user account. If this option is set, password complexity is validated using an external function or internal stored procedure.
5. **Expiry Date –** Password last expiration date and time.
6. **Create Date –** User account creation date and time.
7. **Account Last Lock Date –** User account last lock date and time.
    **Tip:** Do not confuse lock date with expiry date. Oracle supports separate account enabled and account password expired states. The lock date indicates date and time the account was last disabled.
8. **Profile Name –** User resource profile name associated with the account. The profile settings control password expiration and validation policies.
9. **Max Failed Login Attempts –** Number of failed login attempts before lockout of the account.

10. **Password Reuse Times –** Number of days before a password can be reused.
11. **Password Grace Period –** Grace period in days for changing the password after the first successful login after the password has expired.
12. **Password Lock Time –** Number of days the account is locked after the specified number of failed login attempts.
13. **Password Life Time –** Lifetime of the password in days after which the password expires.
14. **Password Verify Function –** User-defined function that performs a password complexity check before a password is assigned.

 **ASE**: This report is available for versions 11.5 and up.

Requirement: None

The following columns are displayed on the report:

1. **Login Name –** The login name.
2. **User External Name –** The external user name, usually full user name or operation system user name, as specified in the user account settings.
3. **Is Login Audited? –** This option indicates whether system auditing is enabled for the login.
4. **Password Change Date –** Date and time of the last password change.
5. **Password Expiry Date –** Password last expiration date and time.
6. **Failed Login Count-** The number of failed logins since the last successful login.

 **DB2:** This report is not available for DB2 because DB2 does not maintain internal database users and their security settings. DB2 relies on operating system security functions for creating and maintaining users.

 **MySQL:** This report is not available for MySQL because the database server does not currently handle the password expiration.

Users with Non-expiring Passwords

To run the report, select the **Reports > Compliance Reports > Users with Non-expiring Passwords** menu.

The report lists all database users (or logins) having non-expiring passwords.

 **Important Notes:**

- The scope of this report is limited to internal database users and logins only. It does not cover externally authenticated users connecting to the database server using

various types of "integrated security" and "enterprise login" methods.

- The report output differs for different database systems. Specifics for each database system are described below.

SQL Server: This report is available only for versions 2005 and up.

Requirement: None

The following columns are displayed on the report:

1. **Login Name –** The login name.
2. **Last Change Date –** The date and time accounts settings were last changed.
   Tip: Password changes are counted along with all other types of changes but the Last Change Date value does not necessarily indicate last password change time.
3. **Authentication Type –** The method used by SQL Server to authenticate user login.
4. **Password Expiration –** The password expiration option for the user account. This value should always display “Yes” for all non-expiring accounts appearing on the report.
5. **Password Complexity –** The password complexity option for the user account. If this option is set, the password complexity is validated using user-defined method.

Oracle: This report is available for versions 7.3 and up.

Requirement: None

Notes: In Oracle, each user account is associated with a resource profile. The profile settings control password expiration and validation policies for associated user accounts. Multiple user accounts can be associated with a single profile.

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **User External Name –** The external user name, usually full user name or operating system user name as specified in the user account settings.
3. **Password Expiration –** The password expiration option for the user account. This option should appear disabled for all user accounts displayed on the report.
4. **Password Complexity –** The password complexity option for the user account. If this option is set, the password complexity is validated using an external function or internal stored procedure.
5. **Create Date –** User account creation date and time.
6. **Profile Name –** User resource profile name associated with the account. The profile settings control password expiration and validation policies.
7. **Account Last Lock Date –** User account last lock date and time.
   Tip: Do not confuse lock date with expiration date. Oracle supports separate account enabled and account password expired states. The lock date indicates date and time the account was last disabled.
8. **Max Failed Login Attempts –** Number of failed login attempts before lockout of the account.

9. **Password Reuse Times –** Number of days before a password can be reused.
10. **Password Grace Period –** Grace period in days for changing the password after the first successful login after the password has expired.
11. **Password Lock Time –** Number of days the account is locked after the specified number of failed login attempts.
12. **Password Life Time –** Lifetime of the password in days before the password expires.
13. **Password Verify Function –** User-defined function that performs a password complexity check before a password is assigned.

 **ASE**: This report is available for versions 11.5 and up.

Requirement: None

The following columns are displayed on the report:

1. **Login Name –** The login name.
2. **User External Name –** The external user name, usually full user name or operating system user name, as specified in the user account settings.
3. **Is Login Audited? –** This option indicates whether system auditing is enabled for the login.
4. **Password Change Date –** Date and time of the last password change.
5. **Password Expiry Date –** Password last expiration date and time. If the account was assigned an expiration date before the last account change, this value indicates last expiration date before the change.
6. **Failed Login Count-** The number of failed logins since the last successful login.

 **DB2:** This report is not available for DB2 because DB2 does not maintain internal database users and their security settings. DB2 relies on operating system security functions for creating and maintaining users.

 **MySQL:** This report is not available for MySQL because the database server does not currently handle the password expiration.

Users Having Administrative Privileges

To run the report, select the **Reports > Compliance Reports > Users Having Administrative Privileges** menu.

The report lists all database users (or logins) having administrative privileges.

 **Notes:** The report output differs for different database systems. Specifics for each database system are described below.

 **SQL Server:** This report is available for version 2000 and up.

Requirement: None

The following columns are displayed on the report:

1. **Login Name –** The login name.
2. **System Admin –** Indicates whether the account has been assigned System Administrator functional role.
3. **Security Admin –** Indicates whether the account has been assigned Security Administrator functional role.
4. **Server Admin –** Indicates whether the account has been assigned Server Administrator functional role.
5. **Setup Admin –** Indicates whether the account has been assigned Setup Administrator functional role.
6. **Process Admin –** Indicates whether the account has been assigned Process Administrator functional role.
7. **Disk Admin –** Indicates whether the account has been assigned Disk Administrator functional role.
8. **Database Creator –** Indicates whether the account has been assigned **Disk Administrator** functional role.

 **Oracle**: This report is available for versions 7.3 and up.

Requirement: None

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **User External Name –** The external user name, usually the full user name or operating system user name, as specified in the user account settings.
3. **Create Date** – User account creation date and time.
4. **Account Status** – The current account status.
5. **Has DBA-type Role? –** Indicates whether or not the account has been assigned the built-in DBA role.
6. **Can Alter Any Table? –** Indicates whether the account was assigned the ALTER ANY TABLE privilege directly or indirectly through other roles.
7. **Can Create Any Table? –** Indicates whether the account was assigned the CREATE ANY TABLE privilege directly or indirectly through other roles.
8. **Can Drop Any Table? –** Indicates whether the account was assigned the DROP ANY TABLE privilege directly or indirectly through other roles.

ASE: This report is available for versions 11.5 and up.

Requirement: None

The following columns are displayed on the report:

1. **Login Name –** The user login name.
2. **User External Name –** The external login name, usually the full user name or operation system user name, as specified in the user account settings.
3. **Account Status** – The current account status.

DB2: This report is available only for DB2 UDB for Linux, Unix and Windows versions 7.1 and up. Some columns on the report are only visible for DB2 9.0 and up. See the following report column descriptions for more details.

Requirement: None

Notes: DB2 does not maintain internal database users and their security settings. DB2 relies on operating system security functions for creating and maintaining users and user group memberships. This report is limited to administrative users who have been explicitly granted admin privileges or who have granted such privileges to other users and roles.

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **DBA Authority –** Indicates whether the account has been assigned the functional DBA role.
3. **Create Table Authority –** Indicates whether the account has been assigned the Create Table privileges.
4. **Connect Authority –** Indicates whether the account has been assigned the Connect privileges.
5. **Bind Authority –** Indicates whether the account has been assigned the Bind privileges.
6. **Non-Fenced Procs Authority –** Indicates whether the account has been assigned privileges for creating Non-fenced procedures.
7. **Load Authority –** Indicates whether the account has been assigned the Load privileges.
8. **External Procs Authority –** Indicates whether the account has been assigned the External Procs privileges. This column only appears on reports run against DB2 9.0 or later.
9. **Library Admin Authority –** Indicates whether the account has been assigned the Library Admin privileges. This column only appears on reports run against DB2 9.0 or later.
10. **Security Admin Authority –** Indicates whether the account has been assigned the Security Admin privileges. This column only appears on reports run against DB2 9.0 or later.

MySQL: This report is available only for MySQL versions 5.0 and up.

Requirement: None

The following columns are displayed on the report:

1. **User Name –** The database user name.
2. **Kill Sessions –** Indicates whether the account has been granted the Kill Sessions system privilege.
3. **Create Users –** Indicates whether the account has been granted the Create Users system privilege.
4. **Shutdown Server –** Indicates whether the account has been granted the Shutdown Server system privilege.
5. **Reload Sys. Changes –** Indicates whether the account has been granted the Flush and Reload System and Security changes system privilege.
6. **List Processes –** Indicates whether the account has been granted the List Processes system privilege.
7. **Read/Write Files –** Indicates whether the account has been granted the Read/Write Files system privilege.
8. **Create Databases –** Indicates whether the account has been granted the Create Database Schemas system privilege.

Recent Administrator Logins

To run the report, select the **Reports > Compliance Reports > Recent Administrator Logins** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report lists all logins within the selected reporting period using accounts with administrative privileges.

SQL Server: This report is available for versions 2000 and up.

Requirements:

- DB Audit system auditing must be installed
- Login (successful) event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Time –** The date and time of the login.
2. **Login Name –** The login name.
3. **Terminal -** Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.

4. **OS User Name –** Network user name of the database user.
5. **Program -** The name of the program from which the connection was made.

 **Oracle**: This report is available for versions 7.3 and up.

Requirements:

- System auditing must be enabled
- Connect or Session auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Time** – The date and time of the login.
2. **User Name –** The database user name.
3. **OS User Name –** Network user name of the database user.
4. **Terminal** - Name of the network terminal or user workstation from which the database connection was made.

 **ASE:** This report is available only for versions 11.5 and up.

Requirements:

- System auditing must be installed
- LOGINS auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Time –** The date and time of the login.
2. **Login Name –** The login name.
3. **User External Name** – The external login name, usually full user name or operation system user name, as specified in the user account settings.

 **DB2**: This report is available only for DB2 UDB for Linux, Unix and Windows versions 7.1 and up.

Notes: DB2 does not maintain internal database users and their security settings. DB2 relies on operating system security functions for creating and maintaining users and user group memberships. This report is limited to administrative users who have explicitly granted admin privileges or who have granted such privileges to other users and roles.

Requirements:

- DB Audit system auditing must be installed
- User Authentication auditing must be enabled

- Authorization Checking auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Time** – The date and time of the login.
2. **User Name –** The database user name.
3. **OS User Name –** Network user name of the database user.
4. **Terminal** - Name of the network terminal or user workstation from which the database connection was made. For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
5. **Program** - The name of the program from which the connection was made.

 **MySQL:** This report is available for versions 5.0.2 and up.

Requirements:

- DB Audit system auditing must be installed
- Login (successful) event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Time –** The date and time of the login.
2. **User Name –** The database user name.
3. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.

Recent Privileged Operations (Create, Drop, Alter)

To run the report, select the **Reports > Compliance Reports > Recent Privileged Operations (Create, Drop, Alter)** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default, the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report lists all recently executed CREATE, DROP, ALTER or RENAME object-level operations.

 **Tip:** For all other types of privileged operations such as DBCC, BACKUP, RESTORE or ALTER SESSION, you can use the available system auditing reports. For more information, see the System Audit and Security Reports topic in CHAPTER 7. To see the text of executed SQL Queries, use the Text of SQL Queries Reports. For more about this report, see Text of SQL Queries Report topic.

Notes:

- The report displays successful privileged operations only. To review failed operations, see the Database Errors and Exceptions report available in the System Audit Reports section. For more information about this report, read Database Errors Report topic.
- The report output differs for different database systems. Specifics for each database system are described below.

SQL Server: This report is available only for versions 2000 and up.

Requirements:

- DB Audit system auditing must be installed
- Privileged Operation (CREATE/DROP/etc...) auditing must be enabled
- Schema Object Access auditing must be enabled
- Schema Object Derived Permission auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Name –** The login name of the user who executed the operation.
2. **OS User Name –** Network user name of the database user.
3. **Event Time –** The date and time when the operation was executed.
4. **Action –** The type of executed operation.
5. **Database –** The name of the database on which the operation was executed.
6. **Object Type –** The type of object affected by the operation.
7. **Object Name –** The full name of the object affected by the operation, including schema name, if applicable
8. **Terminal -** Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
9. **Program -** The name of the program used to execute the operation.
10. **SQL Command –** The actual SQL command executed on the server.

Oracle: This report is available for versions 7.3 and up.

Requirements:

- System auditing must be enabled
- All CREATE, DROP and ALTER statement-level operations must be enabled. Note that enabling object level-only auditing will not allow you to have the complete report reflecting all changes in the database.
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **User Name –** The name of the database user who to execute the operation.
2. **OS User Name –** Network user name of the database user.
3. **Event Time –** The date and time when the operation was executed.
4. **Action –** The type of operation attempted.
5. **Object Name –** The full name the database object affected.
6. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.

 **ASE**: This report is available only for versions 11.5 and up.

Requirements:

- System auditing must be installed
- CREATE, DROP and ALTER auditing must be enabled for each monitored database
- SQL Text auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **Login Name –** The login name of the user who executed the operation.
2. **Event Time –** The date and time when the operation was executed.
3. **Action –** The type of operation attempted.
4. **Database –** The name of the database on which the operation was executed.
5. **Schema –** Schema name of the database object affected.
6. **Object Name -** Name of the database object affected.
7. **Additional Info –** The text of the SQL command used to modify the object or privileges used for the operation.

 **DB2:** This report is available only for DB2 UDB for Linux, Unix and Windows versions 7.1 and up.

Requirements:

- DB Audit system auditing must be installed
- Object Drop and Create auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **User Name –** The name of the database user who executed the operation.
2. **Event Time –** The date and time when the operation was executed.
3. **Action –** The type of operation executed.
4. **Database –** The name of the database on which the operation was executed.

5. **Schema –** Schema name of the database object affected.
6. **Object Name -** Name of the database object affected.
7. **Privilege –** Description of the privilege being granted or revoked.
8. **Terminal -** Name of the network terminal or user workstation from which the database connection was made. For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID including connection handles, as reported by DB2 audit.
9. **Program -** The name of the program used to execute the operation.

 **MySQL:** This report is available only for MySQL versions 5.0.2 and up.

Requirements:

- DB Audit system auditing must be installed
- Object Create/Drop/Alter/Rename/Truncate auditing must be enabled
- Auditing of INSERT, DELETE, UPDATE operations for system tables in MYSQL database schema is highly recommended
- Data must be available in the audit trail for the chosen reporting period

The following columns are displayed on the report:

1. **User Name –** The name of the database user who executed the operation.
2. **Event Time –** The date and time when the operation was executed.
3. **Action –** The type of operation executed.
4. **Object Type –** Type of the database object affected.
5. **Schema –** Schema name of the database object affected.
6. **Object Name -** Name of the database object affected.
7. **Privilege –** Description of the privilege being granted or revoked.
8. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.
9. **SQL Command –** First 255 characters of the SQL command .

Behavioral Analysis Reports

Behavioral reports analyze data in the system audit trail and search for patterns of suspicious activities enabling you to identify various security violations that normally do not trigger alerts in other database and network security systems. These reports allow you to evaluate activities of database users and ensure that you have adequate database security settings in place. They also allow you to identify poorly protected systems and poorly written applications that might provide users with unauthorized access to your data and files.

Suspicious Connections from Untrusted Domains

To run the report, select the **Reports > Behavioral Analysis > Suspicious Logons from Untrusted Domains** menu. The **Specify Report Filter** dialog will appear. Use the filter to enter your network domain name and to select the desired reporting period. By default, the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays successful and failed database server login attempts performed by users authenticated within a network domain whose name differs from the trusted domain name you specify in report parameters.

SQL Server: The report is limited to users logging in to SQL Server using Windows Integrated Authentication option, the so-called Trusted Connection. The report does not cover users using SQL Server based authentication options. This report is available for SQL Server versions 2000 and later.

Oracle: This report is limited to users logging in to the database using various networking protocols. It doesn't cover users logging in locally using shared memory and other local IPC methods. The report is available for all Oracle versions.

DB2: This report is limited to users logging in to the database using client-side authentication. The report does not cover users using server-side authentication. The report is available for DB2 UDB versions 8.0 and later.

MySQL, ASE, ASA: This report is not currently supported.

Requirements (all database systems):

- DB Audit system auditing must be installed
- Both Login (failed) event and Login (successful) event auditing should be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Login Name –** The login name of the user who attempted to make a connection.
2. **OS User Name –** Network user name of the database user. The name is provided in **domain \ user** format.
3. **Terminal –** Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
4. **Login Time –** The date and time of the attempted login.
5. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
6. **Duration**– Duration of the database session in case of a successful login.
7. **Status –** Indicates whether the login succeeded or failed.

Oracle, DB2: The following columns are displayed on the report:

1. **User Name –** The name of the database user who attempted to make a connection.
2. **OS User Name –** Network user name of the database user. The name is provided in

domain \ user format.

3. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made. For DB2: for TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
4. **Login Time –** The date and time of the attempted login.
5. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
6. **Duration**– Duration of the database session in case of a successful login.
7. **Status –** This indicates whether the login succeeded or failed.

Recurring Logon Failures

To run the report, select the **Reports > Behavioral Analysis > Recurring Logon Failures** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays consecutive failed attempts to connect to the server that occurred on the same day during the selected reporting period.

 **ASA**: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Both Login (failed) and Login (successful) event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

 **SQL Server:** The following columns are displayed on the report:

1. **Date –** The date when attempts were made to connect to the database server.
2. **Login Name –** The login name of the user who attempted to make a connection.
3. **OS User Name –** Network user name of the database user. The name is provided in **domain \ user** format.
4. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connections were attempted.
5. **First Time –** The time of the first failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
6. **Last Time –** The time of the last failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
7. **Total Attempts –** The total number of consecutive failed connection attempts made by the user on the specified date.

ASE: The following columns are displayed on the report:

1. **Date –** The date when attempts were made to connect to the database server.
2. **Login Name –** The login name of the user who attempted to make a connection.
3. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connections were attempted (this value is not available for all ASE versions and communication protocols)
4. **First Time –** The time of the first failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
5. **Last Time –** The time of the last failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
6. **Total Attempts –** The total number of all consecutive failed connection attempts made by the user on the specified date.

Oracle, DB2: The following columns are displayed on the report:

8. **Date –** The date when attempts were made to connect to the database server.
9. **User Name –** The name of the database user who attempted to make a connection.
1. **OS User Name –** Network user name of the database user. The name is provided in **domain \ user** format.
2. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connections were attempted. For DB2: for TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
3. **First Time –** The time of the first failed connection attempt for the date specified in the Date column and user specified in the Login Name column.
4. **Last Time –** The time of the last failed connection attempt for the date specified in the Date column and user specified in the Login Name column.
5. **Total Attempts –** The total number of all consecutive failed connection attempts made by the user on the specified date.

MySQL: The following columns are displayed on the report:

1. **Date –** The date when attempts were made to connect to the database server.
2. **User Name –** The name of the database user who attempted to make a connection.
3. **Terminal –** Name of the network terminal or user workstation from which the database connections were attempted.
4. **First Time –** The time of the first failed connection attempt for the date specified in the Date column and user specified in the Login Name column.
5. **Last Time –** The time of the last failed connection attempt for the date specified in the Date column and user specified in the Login Name column.
6. **Total Attempts –** The total number of all consecutive failed connection attempts made by the user on the specified date.

Local Logons

To run the report, select the **Reports/Behavioral Analysis > Local Logons** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays all attempted local connections to the server during the chosen reporting period along with the connection methods used.

ASE, ASA: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Both Login (failed) event and Login (successful) event auditing should be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Connect Method –** The local connection method, one of the following:
   - TCP/IP
   - Local Pipes
   - Shared Memory
   - Other
2. **Login Name –** The login name of the user who attempted to make a connection.
3. **OS User Name –** Network user name of the database user.
4. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
5. **Login Time –** The date and time of the attempted login.
6. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
7. **Duration –** Duration of the database session in case of a successful login.
8. **Status –** This indicates whether the login succeeded or failed.

Oracle, DB2: The following columns are displayed on the report:

1. **Connect Method –** The local connection method, one of the following:
   - TCP/IP
   - Local Pipes
   - Shared Memory

- Other

2. **User Name –** The name of the database user who attempted to make a connection.
3. **OS User Name –** Network user name of the database user.
4. **Terminal –** Name of the network terminal or user workstation from which the database connection was made. For DB2: for TCP/IP connections, the Terminal field displays the IP address of the user's terminal. for local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
5. **Login Time –** The date and time of the attempted login.
6. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
7. **Duration –** Duration of the database session in case of a successful login.
8. **Status –** Indicates whether the login succeeded or failed.

 **MySQL:** The following columns are displayed on the report:

1. **User Name –** The name of the database user who attempted to make a connection.
2. **OS User Name –** The full database user name in the **dbuser@host** format. This name should normally match user's network name.
3. **Terminal –** Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
4. **Login Time –** The date and time of the attempted login.
5. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
6. **Duration –** Duration of the database session in case of a successful login.
7. **Status –** This indicates whether the login succeeded or failed.

Multi-user Failed Logon Attempts from Same Terminal

To run the report, select the **Reports > Behavioral Analysis > Multi-user Logon Attempts from Same Terminal** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays consecutive failed logons attempted from the same network terminal using different database user names during the selected reporting period.

 **ASA**: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Login (failed) event auditing must be enabled

- Data must be available in the audit trail for the chosen reporting period

SQL Server, ASE: The following columns are displayed on the report:

1. **Date –** The date when attempts were made to connect to the database server.
2. **Terminal –** The TCP/IP address or name of network terminal or user workstation from which the database connections were attempted. This is the value reported by the Host connection parameter.
3. **Login Names –** Two or more login names used to make connections from the terminal specified in the Terminal column.
4. **First Time –** The time of the first failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
5. **Last Time –** The time of the last failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
6. **Total Attempts –** The total number of connection attempts made from the specified terminal on the specified date.

Oracle, DB2, MySQL: The following columns are displayed on the report:

1. **Date –** The date when attempts were made to connect to the database server.
2. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connections were attempted. This is the value reported by the Host connection parameter.
3. **User Names –** Two or more database user names used to make connections from the terminal specified in the Terminal column.
4. **First Time –** The time of the first failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
5. **Last Time –** The time of the last failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.
6. **Total Attempts –** The total number of connection attempts made from the specified terminal on the specified date.

Database Login Sharing

To run the report, select the **Reports > Behavioral Analysis > Database Login Sharing** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays users and terminals that share a common database login to connect to the database server. The report scope is limited to the selected reporting period.

ASE, ASA: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Login (successful) event auditing must be enabled.
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed in the details:

1. **Date –** The date when attempts were made to connect to the database server.
2. **Login Name –** The login name of the user who attempted to make a connection.
3. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connections were attempted. This is the value reported by the Host connection parameter.
4. **First Logon Time –** The time of the first failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.

Oracle, DB2: The following columns are displayed in the details:

1. **Date –** The date when attempts were made to connect to the database server.
2. **User Name –** The name of the database user who attempted to make a connection.
3. **OS User Name –** Network user name of a network connection used to connect to the database.
4. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connections were attempted. For DB2: for TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
5. **First Logon Time –** The time of the first failed connection attempt for the date specified in the Date column and the user specified in the Login Name column.

MySQL: The following columns are displayed in the details:

1. **Date –** The date when attempts were made to connect to the database server.
2. **User Name –** The name of the database user who attempted to make a connection.
3. **Terminal –** Name of network terminal or user workstation from which the database connections were attempted. This is the value reported by the Host connection parameter.
4. **First Logon Time –** The time of the first failed connection attempt for the date specified in the Date column and user specified in the Login Name column.

New User Accounts and Database Connections

To run the report, select the **Reports > Behavioral Analysis > New User Accounts and Database Connections** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays connections made by users who have never previously connected to the database server during the selected reporting period. The report also shows events related to new database account creation, such as creation of a new user or new login. These events can be correlated in order to pinpoint unauthorized database logins.

Important Notes: Users authorized to log in to SQL Server database servers using their Windows group membership may appear as new users without directly granted login permissions. Using this report, you can find users who gained access to the database server by illegally joining authorized Windows user groups.

The report visibility is limited to events available in the audit trail. If you employ audit trail purging procedures, the report might be unable to find all related events and might report some old users as new because their logins had been created prior to the first event time available in the audit trail.

ASE, ASA: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Login (successful) event auditing must be enabled
- Security Changes event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Connect Method –** The connection method, one of the following:
   - TCP/IP
   - Local Pipes
   - Shared Memory
   - Other
   - [Blank] for Failed Connections and for Login Created event records displayed on the report.
2. **Login Name –** The login name of the user who attempted to make a connection.
3. **OS User Name –** Network user name of the database user.
4. **Terminal –** The TCP/IP address or Name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
5. **Login Time –** The date and time of the attempted login.
6. **Logout Time –** The date and time when the logout occurred in case of previous successful login.

7. **Duration –** Duration of the database session in case of a successful login.
8. **Status –** This indicates whether the login succeeded or failed.

Oracle, DB2: The following columns are displayed on the report:

1. **Connect Method –** The connection method, one of the following:
   - TCP/IP
   - Local Pipes
   - Shared Memory
   - Other
   - [Blank] for Failed Connections and for Login Created event records displayed on the report.
2. **User Name –** The name of the database user who attempted to make a connection.
3. **OS User Name –** Network user name of the database user.
4. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made. For DB2: For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
5. **Login Time –** The date and time of the attempted login.
6. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
7. **Duration –** Duration of the database session in case of a successful login.
8. **Status –** Indicates whether the login succeeded or failed.

MySQL: The following columns are displayed on the report:

1. **User Name –** The name of the database user who attempted to make a connection.
2. **Terminal –** Name of the network terminal or user workstation from which the database connection was made.
3. **Login Time –** The date and time of the attempted login.
4. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
5. **Duration –** Duration of the database session in case of a successful login.
6. **Status –** This indicates whether the login succeeded or failed.

New Terminals Used for Database Connections

To run the report, select the **Reports > Behavioral Analysis > New Terminals Used for Database Connections** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays connections made from network terminals and workstations that have never been used to connect to the database server during the selected reporting period. The report also shows events related to new database account creation, such as creation of a new user or new login. Account creation and first login events can be correlated for the purpose of finding and investigating unauthorized database logins.

Important Notes: Report visibility is limited to events available in the audit trail. If you employ audit trail purging procedures, the report might be unable to find all related events and might report some old "known" terminal as new.

ASE, ASA: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Both Login (failed) and Login (successful) event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Connect Method –** The connection method, one of the following:
   - TCP/IP
   - Local Pipes
   - Shared Memory
   - Other
   - [Blank] for Failed Connections and for Login Created event record displayed on the report.
2. **Login Name –** The login name of the user who attempted to make a connection.
3. **OS User Name –** Network user name of the database user.
4. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made. This is the value reported by the Host connection parameter.
5. **Login Time –** The date and time of the attempted login.
6. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
7. **Duration –** Duration of the database session in case of a successful login.
8. **Status –** This indicates whether the login succeeded or failed.

Oracle, DB2: The following columns are displayed on the report:

1. **Connect Method –** The connection method, one of the following:
   - TCP/IP
   - Local Pipes
   - Shared Memory
   - Other
   - [Blank] for Failed Connections and for Login Created event record displayed on the report.
2. **User Name –** The name of the database user who attempted to make a connection.
3. **OS User Name –** Network user name of the database user.
4. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made. For DB2: For TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID< including connection handles, as reported by DB2 audit.
5. **Login Time –** The date and time of the attempted login.
6. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
7. **Duration –** Duration of the database session in case of a successful login.
8. **Status –** This indicates whether the login succeeded or failed.

MySQL: The following columns are displayed on the report:

1. **User Name –** The name of the database user who attempted to make a connection.
2. **Terminal –** Name of the network terminal or user workstation from which the database connection was made.
3. **Login Time –** The date and time of the attempted login.
4. **Logout Time –** The date and time when the logout occurred in case of previous successful login.
5. **Duration –** Duration of the database session in case of a successful login.
6. **Status –** This indicates whether the login succeeded or failed.

Activities of Users Prior to Termination

To run the report, select the **Reports > Behavioral Analysis > Activities of Users Prior to Termination** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during previous two months (60 or 61 days depending on the calendar month).

The report displays daily summary statistics for new activities of users whose accounts have been terminated. Report scope is limited to the selected reporting period. Only users terminated during the selected period are reported.

ASE, ASA: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Security Changes event auditing must be enabled
- Schema Object Access event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Login Name –** The login name of the user whose activities are reported.
2. **Day –** The date of the activity.
3. **Database –** The name of the database where the activity took place.
4. **Schema –** The name of the database schema owning the object accessed by the user.
5. **Object Name –** The name of the accessed database object.
6. **Total** – The total number of accesses during the day specified in the Day column.
7. **Ins –** The total number of INSERT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
8. **Del –** The total number of DELETE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
9. **Upd –** The total number of UPDATE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
10. **Sel –** The total number of SELECT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
11. **Exec –** The total number of EXECUTE operations executed against the object during the day specified in the Day column. This value is only applicable to database stored procedures and user-defined functions.

Oracle, DB2, MySQL: The following columns are displayed on the report:

1. **User Name –** The name of the database user whose activities are reported.
2. **Day –** The date of the activity.
3. **Schema –** The name of the database schema owning the object accessed by the user.
4. **Object Name –** The name of the accessed database object.
5. **Total** – The total number of accesses during the day specified in the Day column.
6. **Ins –** The total number of INSERT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
7. **Del –** The total number of DELETE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.

8. **Upd –** The total number of UPDATE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
9. **Sel –** The total number of SELECT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
10. **Exec –** The total number of CALL operations executed against the object during the day specified in the Day column. This value is only applicable to database stored procedures and user-defined functions.

Suspicious Data Access Never Done Before

To run the report, select the **Reports > Behavioral Analysis > Suspicious Data Access Never Done Before** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays daily summary statistics for new activities of users who are not known to have performed such activities previously. The report scope is limited to the selected reporting period. Only activities recorded during the selected period are reported, although the search of past activities is performed against all data available in the audit trail.

 **Important Notes:** Report visibility is limited to events available in the audit trail. If you employ audit trail purging procedures, the report might be unable to find all related events and might report some activities as new if they previously occurred only before the first event time available in the audit trail.

 **ASE, ASA**: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Schema Object Access event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

 **SQL Server:** The following columns are displayed on the report:

1. **Login Name –** The login name of the user whose activities are reported.
2. **Day –** The date of the activity.
3. **Database –** The name of the database where the activity took place.
4. **Schema –** The name of the database schema owning the object accessed by the user.
5. **Object Name –** The name of the accessed database object.
6. **Total** – The total number of accesses during the day specified in the Day column.
7. **Ins –** The total number of INSERT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
8. **Del –** The total number of DELETE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.

9. **Upd –** The total number of UPDATE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
10. **Sel –** The total number of SELECT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
11. **Exec –** The total number of EXECUTE operations executed against the object during the day specified in the Day column. This value is only applicable to database stored procedures and user-defined functions.

 **Oracle, DB2, MySQL:** The following columns are displayed on the report:

1. **User Name –** The name of the database user whose activities are reported.
2. **Day –** The date of the activity.
3. **Database –** The name of the database where the activity took place.
4. **Schema –** The name of the database schema owning the object accessed by the user.
5. **Object Name –** The name of the accessed database object.
6. **Total** – The total number of accesses during the day specified in the Day column.
7. **Ins –** The total number of INSERT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
8. **Del –** The total number of DELETE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
9. **Upd –** The total number of UPDATE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
10. **Sel –** The total number of SELECT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
11. **Exec –** The total number of CALL operations executed against the object during the day specified in the Day column. This value is only applicable to database stored procedures and user-defined functions.

After Business Hours Data Access

To run the report, select the **Reports > Behavioral Analysis > After Business Hours Data Access** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays hourly summary statistics for activities of users after regular business hours. The report scope is limited to events occurring before 8:00 AM and after 6:00 PM. The overall report scope is limited to the selected reporting period. Only activities recorded during the selected period are reported

ASE, ASA: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Schema Object Access event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Login Name –** The login name of the user whose activities are reported.
2. **Day –** The date of the activity.
3. **Hour –** The time of the activity expressed as the beginning of the hour in 24-hour time format. For example, for all events occurred between 7:00 PM and 7:59 PM, the Hour value is displayed as 19:00.
4. **Database –** The name of the database on which the activity took place.
5. **Schema –** The name of the database schema owning the object accessed by the user.
6. **Object Name –** The name of the accessed database object.
7. **Total** – The total number of accesses during the day specified in the Day column.
8. **Ins –** The total number of INSERT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
9. **Del –** The total number of DELETE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
10. **Upd –** The total number of UPDATE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
11. **Sel –** The total number of SELECT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
12. **Exec –** The total number of EXECUTE operations executed against the object during the day specified in the Day column. This value is only applicable to database stored procedures and user-defined functions.

Oracle, DB2, MySQL: The following columns are displayed on the report:

1. **User Name –** The name of the database user whose activities are reported.
2. **Day –** The date of the activity.
3. **Hour –** The time of the activity expressed as the beginning of the hour in 24-hour time format. For example, for all events occurring between 7:00 PM and 7:59 PM, the Hour value is displayed as 19:00.
4. **Database –** The name of the database on which the activity took place.
5. **Schema –** The name of the database schema owning the object accessed by the user.
6. **Object Name –** The name of the accessed database object.
7. **Total** – The total number of accesses during the day specified in the Day column.
8. **Ins –** The total number of INSERT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.

9. **Del –** The total number of DELETE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
10. **Upd –** The total number of UPDATE operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
11. **Sel –** The total number of SELECT operations executed against the object during the day specified in the Day column. This value is only applicable to database tables and views.
12. **Exec –** The total number of CALL operations executed against the object during the day specified in the Day column. This value is only applicable to database stored procedures and user-defined functions.

Unusually High Activity (Twice Above Average)

To run the report, select the **Reports > Behavioral Analysis > Unusually High Activity** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays hourly summary statistics for activities of users who generated at least twice the usual number of audit events as they normally generate using the same database applications at the same hour of day. Report scope is limited to the selected reporting period. Only activities recorded during the selected period are reported. To calculate hourly averages, DB Audit analyzes all data available in the audit trail.

 **ASE, ASA**: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- At the very minimum, Schema Object Access event auditing must be enabled. It is recommended that all available event types be enabled for auditing in order for the report to produce accurate results.
- Data must be available in the audit trail for the chosen reporting period

 **SQL Server:** The following columns are displayed on the report:

1. **Event Time –** The time of the event rounded to the beginning of the hour.
2. **Login Name –** The login name of the user whose activities are reported.
3. **OS User Name –** Network user name of the database user.
4. **Program –** The name of the program from which the user attempted to execute the operation.
5. **Event Count –** The total number of events recorded during the hour specified in the Event Time column for the user and application specified in the Login Name and Program Name columns.
6. **Hourly Average–** The average number of events typically registered for the user and

application specified in the Login Name and Program Name columns for the hour specified in the Event Time column.

Oracle: The following columns are displayed on the report:

1. **Event Time –** The time of the event rounded to the beginning of the hour.
2. **User Name –** The name of the database user whose activities are reported.
3. **OS User Name –** Network user name of the database user.
4. **Program –** Column reserved for future use.
5. **Event Count –** The total number of events recorded during the hour specified in the Event Time column for the user specified in the User Name column.
6. **Hourly Average–** The average number of events typically registered for the user specified in the User Name column for the hour specified in the Event Time column.

DB2: The following columns are displayed on the report:

7. **Event Time –** The time of the event rounded to the beginning of the hour.
8. **User Name –** The name of the database user whose activities are reported.
9. **OS User Name –** Network user name of the database user.
10. **Program –** The name of the program from which the user attempted to execute the operation.
11. **Event Count –** The total number of events recorded during the hour specified in the Event Time column for the user and application specified in the User Name and Program Name columns.
12. **Hourly Average–** The average number of events typically registered for the user and application specified in User Name and Program Name columns for the hour specified in the Event Time column..

MySQL: The following columns are displayed on the report:

1. **Event Time –** The time of the event rounded to the beginning of the hour.
2. **User Name –** The name of the database user whose activities are reported.
3. **OS User Name –** Network user name of the database user.
4. **Program –** Column reserved for future use.
5. **Event Count –** The total number of events recorded during the hour specified in the Event Time column for the user specified in the User Name column.
6. **Hourly Average–** The average number of events typically registered for the user specified in the User Name column for the hour specified in the Event Time column.

Audit Trail Configuration and Data Changes

To run the report, select the **Reports > Behavioral Analysis > Audit Trail Configuration and Data Changes** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default, the report is limited to events occurred during previous two months (60 or 61 days depending on the calendar month).

The report displays activities of users who attempted to make changes in the audit system configuration or attempted to delete or modify data in the local adult trail tables during the selected reporting period.

ASE, ASA: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Schema Object Access event auditing must be enabled
- Security Changes event auditing must be enabled
- Schema Changes event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **OS User Name –** Network user name of the database user.
2. **Login Name –** The login name of the user whose activities are reported.
3. **Terminal -** Name of the network terminal or user workstation from which the database connection was made as specified in the Host connection property.
4. **Event Time –** The time of the event.
5. **Database –** The name of the database in which the activity occurred or attempted.
6. **Action Name –** The type of the operation attempted.
7. **Object Type –** The type of the affected database object.
8. **Object Name –** The full name of the affected database object, including schema part.
9. **Program –** The name of the program from which the user attempted to execute the operation.

Oracle, DB2: The following columns are displayed on the report:

1. **OS User Name –** Network user name of the database user.
2. **User Name –** The name of the database user whose activities are reported.
3. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made. For DB2: For TCP/IP connections, the Terminal field displays the IP address of the user's terminal: For local connections, the Terminal field displays [LOCAL]: For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID, including connection handles, as reported by DB2 audit.
4. **Event Time –** The time of the event.
5. **Action Name –** The type of the operation attempted.
6. **Object Type –** The type of the affected database object.
7. **Object Name –** The full name of the affected database object, including schema part.

MySQL: The following columns are displayed on the report:

1. **User Name –** The name of the database user whose activities are reported.
2. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.
3. **Event Time –** The time of the event.
4. **Action Name –** The type of operation attempted.
5. **Object Type –** The type of the affected database object.
6. **Object Name –** The full name of the affected database object, including schema part.

SQL Injection Attempts

To run the report, select the **Reports > Behavioral Analysis > SQL Injection Attempts** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays database activities with patterns common for SQL Injection Attacks. The report scope is limited to the selected reporting period.

Important Notes: SQL Injection Attacks can take many different forms. The number of forms is virtually unlimited so it is not possible, in practice, to accurately detect all forms of SQL Injection Attacks. This report has been designed to find only common forms such as the addition of an extra UNION clause to the end of an existing SQL query or the addition of various elements to the end of the WHERE clause that would invalidate the entire SQL query scope. An example of the latter addition would be the statement "OR 1=1." To reduce the number of potential false positives, this report analyses all events available in the audit trail by performing an extensive search for similar queries used in the past but without the added SQL parts.

Report visibility is limited to events available in the audit trail. If you employ audit trail purging procedures, the report might be unable to find all related events and might miss some SQL injections.

ASE, ASA, DB2: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Login event auditing must be enabled
- Schema Object Access event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Event Time –** The time of the event.
2. **OS User Name –** Network user name of the database user.
3. **Login Name –** The login name of the user whose activities are reported.
4. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.

5. **Program –** The name of the program from which the user attempted to execute the operation.
6. **SQL Command –** The text of the suspect SQL query (first 2000 characters).
7. **Injection Type –** Type of SQL injection, for example, UNION, WHERE.

Oracle: Oracle 10g and later is required. Extended database auditing mode must be enabled in order to capture executed SQL commands and store them in the audit trail.

The following columns are displayed on the report:

1. **Event Time –** The time of the event.
2. **OS User Name –** Network user name of the database user.
3. **User Name –** The login name of the user whose activities are reported.
4. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made.
5. **SQL Command –** The text of the suspect SQL query (first 4000 characters).
6. **Injection Type –** Type of SQL injection, for example, UNION, WHERE.

MySQL: The following columns are displayed on the report:

1. **Event Time –** The time of the event.
2. **User Name –** The login name of the user whose activities are reported.
3. **Terminal –** Name of the network terminal or user workstation from which the database connection was made.
4. **SQL Command –** The text of the suspect SQL query (first 32000 characters).
5. **Injection Type –** Type of SQL injection, for example, UNION, WHERE.

SQL Commands Executing OS Commands

To run the report, select the **Reports > Behavioral Analysis > SQL Commands Executing OS Commands** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays database activities involving execution of operating system commands from SQL queries, such as use of the *sp_cmdtext* system procedure. Report scope is limited to the selected reporting period.

Important Notes: This report covers only known system procedures available by default on the database server. It also looks for execution of user-defined external procedures with common names used for running external processes; for example, *Host*, *Shell* and other frequently used names.

ASE, ASA, DB2: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Schema Object Access event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **OS User Name –** Network user name of the database user.
2. **Login Name –** The login name of the user whose activities are reported.
3. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.
4. **Program –** The name of the program from which the user attempted to execute the operation.
5. **Event Time –** The time of the event.
6. **SQL Query –** The text of the suspect SQL query (first 2000 characters).

Oracle: The following columns are displayed on the report:

1. **OS User Name –** Network user name of the database user.
2. **User Name –** The name of the database user whose activities are reported.
3. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made.
4. **Event Time –** The time of the event.
5. **Schema –** The schema name of the database object used to execute external OS command.
6. **Object name –** The name of the database object used to execute the external OS command.
7. **SQL Query –** The text of the suspect SQL query (first 4000 characters). This column is available for Oracle 10g and up.

MySQL: The following columns are displayed on the report:

1. **User Name –** The name of the database user whose activities are reported.
2. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made.
3. **Event Time –** The time of the event.
4. **Schema –** The schema name of the database object used to execute external OS command.
5. **Object name –** The name of the database object used to execute external OS command.
6. **SQL Query –** The text of the suspect SQL query (first 32000 characters).

Password Changes

To run the report, select the **Reports > Behavioral Analysis > Password Changes** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays password change events occurred during the selected reporting period.

ASE, ASA, DB2, Oracle: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Password Changes event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

SQL Server: The following columns are displayed on the report:

1. **Event Time –** The time of the event.
2. **Target Login/Role Name –** The login name or role name whose password was changed.
3. **Login Name –** The login name of the user who changed the password.
4. **OS User Name –** Network user name of the database user.
5. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.
6. **Program –** The name of the program that was used for the password change.

MySQL: Auditing of INSERT, DELETE, and UPDATE operations for system USERS tables in MYSQL database schema is highly recommended in order to capture direct system table changes.

The following columns are displayed on the report:

1. **Event Time –** The time of the event.
2. **Target User –** The name of the database user whose password was changed.
3. **User Name –** The login name of the user who changed the password.
4. **Terminal -** Name of the network terminal or user workstation from which the database connection was made.

User Privilege Escalation

To run the report, select the **Reports > Behavioral Analysis > User Privilege Escalation** menu.

The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays escalations of user privileges during 24-hour time intervals that occur during the selected reporting period. At least two GRANT or similar commands should be issued for the same login, database user or role within the 24-hour time interval in order for such events to appear on the report.

 **ASE, ASA**: This report is not currently supported.

Requirements:

- DB Audit system auditing must be installed
- Security Changes event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

 **SQL Server:** The following columns are displayed on the report:

1. **Login Name –** The login name of the user who granted new privileges.
2. **Event Time –** The time of the event.
3. **Database –** Name of the database in which the commands were executed.
4. **Action –** The type and scope of security change; for example, GRANT DB PRICIPAL.
5. **Target Object Name –** Full name of the target object for which new permissions were granted. Object name may include object type and schema name.
9. **Target Login/User/Role –** Name of the target login, user, or role who obtained new permissions as the result of the executed command.
6. **Terminal –** Name of the network terminal or user workstation from which the database connection was made.
7. **OS User Name –** Network user name of the database user.
8. **Program –** The name of the program from which the user attempted to execute the operation.
9. **SQL Command –** The text of the SQL query used to grant new permissions (first 2000 characters).

 **Oracle, DB2:** The following columns are displayed on the report:

1. **User Name –** The name of the database user who granted new privileges.
2. **Event Time –** The time of the event.
3. **Action –** The type and scope of security change; for example, EXECUTE IN SCHEMA.
4. **Target Object Name –** Full name of the target object for which new permissions were granted. The object name may include object type and schema name.
5. **Target User/Role –** Name of the target login, user, or role who obtained new permissions as the result of the executed command.
6. **Terminal –** The TCP/IP address or name of the network terminal or user workstation from which the database connection was made. For DB2: for TCP/IP connections, the Terminal field displays the IP address of the user's terminal. For local connections, the Terminal field displays [LOCAL]. For Network Pipe connections, the Terminal field displays [NET PIPE]. For all other connection types, it displays the application ID,

including connection handles, as reported by DB2 audit.

7. **OS User Name –** Network user name of the database user.
8. **Privileges Used –** The privileges available to the grantor that were used to make the reported security changes.

 **MySQL:** The following columns are displayed on the report:

1. **User Name –** The name of the database user who granted new privileges.
2. **Event Time –** The time of the event.
3. **Action –** The type and scope of security change; for example, EXECUTE IN SCHEMA.
4. **Target Object Name –** Full name of the target object for which new permissions were granted. The object name may include object type and schema name.
5. **Target User/Role –** Name of the target login, user, or role who obtained new permissions as the result of the executed command.
6. **Terminal –** Name of the network terminal or user workstation from which the database connection was made.
7. **SQL Command –** The text of the SQL query used to grant new permissions (first 32000 characters).

Statistical Reports

Statistical reports are designed to provide a big picture of database and user activities. Statistical reports reduce large amounts of audit data stored in the system audit trail to meaningful graphical information that is useful for identifying various database security violations and misuses, as well as for pinpointing unusual spikes in activities which are hard to notice on detailed reports.

Logon Activity Charts

To run the report, select the **Reports > Behavioral Analysis > Logon Activity Charts** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays two graphical charts for failed and successful logons as a function of time. It also provides a textual table with daily numbers. The table highlights unusual activities as indicated by daily numbers that significantly exceed daily averages.

 **SQL Server, Oracle, MySQL:** This report is currently available only for SQL Server versions 2000 and later, Oracle versions 8i and later, and MySQL versions 5.0.3 and later.

Requirements:

- DB Audit system auditing must be installed
- Both Login (failed) and Login (successful) event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The following charts are displayed on the report

- Successful logon events as function of time – green chart
- Failed logons events as function of time – red chart

The following columns are displayed in the details:

1. **Day –** Date of activities.
2. **Successful –** Total number of successful logons for the specified date.
3. **Failed –** Total number of successful logons for the specified date
4. **Comments –** Percent of logon attempts as compared to the daily mean and average values.

 **Tip:** Report detail lines highlighted in red indicate spikes in daily logon numbers significantly exceeding the mean value. Comments in these lines describe the excess value as a percent of mean value for the chosen database system and reporting period. All other lines with numbers falling within the normal value range are displayed in black.

User Activity Charts

To run the report, select the **Reports > Behavioral Analysis > User Activity Charts** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report displays a graphical chart for the number of auditing events generated by various user activities as a function of time. It also provides a textual table with daily numbers. The table highlights unusual activities when the daily numbers significantly exceed daily averages.

 **SQL Server, Oracle, MySQL:** This report is currently available only for SQL Server versions 2000 and later, Oracle versions 8i and later, and MySQL versions 5.0.3 and later.

Requirements:

- DB Audit system auditing must be installed
- Schema Object Access event auditing must be enabled
- Data must be available in the audit trail for the chosen reporting period

The report displays a graphical chart of the number of events as a function of time

The following columns are displayed in the details:

1. **Day –** Date of activities.
2. **Event Count –** Total number of audit events for the specified date.
3. **Comments –** Percent of events as compared to the daily mean and average values.

 **Tip:** Report detail lines highlighted in red indicate spikes in daily activity numbers significantly

exceeding the mean value. Comments in these lines describe the excess value as a percent of mean value for the chosen database system and reporting period. All other lines with numbers falling within the normal value range are displayed in black.

Suspicious Gaps in Audit Trail Data

To run the report, select the **Reports > Statistical Reports > Suspicious Gaps in Audit Trail Data** menu. The **Specify Report Filter** dialog will appear. Use the filter to select the desired reporting period. By default the report is limited to events occurred during the previous two months (60 or 61 days depending on the calendar month).

The report performs statistical analysis of the historical data in the system audit trail and displays records indicating suspicious periods of no activity or low activity that contradict historical patterns for that date and time. A graphical chart is also displayed for the five longest recent gaps in the data. The report scope is limited to the selected reporting period.

 **SQL Server:** This report is currently available only for SQL Server versions 2000 and later.

Requirements:

- DB Audit system auditing must be installed
- Schema Object Access event auditing must be enabled
- Logon event auditing is recommended to have.
- Data must be available in the audit trail for the chosen reporting period

The report displays last five significant gaps. The X-axis values indicate the approximate duration of the gap in minutes; the Y-axis values indicate the approximate start date and time of the gap.

The following columns are displayed in the details:

1. **Week Day –** Weekday name.
2. **Start Time –** Approximate start time of the suspicious low activity period.
3. **End Time –** Approximate end time of the suspicious low activity period
4. **Duration –** Approximate duration of the suspicious low activity period. The duration is expressed in *hour:minute* format.
5. **Actual Event Count –** Total number of audit events for the period.
6. **Mean Event Count –** Historical mean value for the number of audit events typically occurring during the period.

Security Snapshots Reports

Security Snapshots reports are designed to provide a point-in-time view of the database security. To use Security Snapshots reports, the Security Snapshots feature must be installed and running.

Enterprise-wide User Directory

To run the report, select **Tools > Security Snapshot** menu and follow the instructions provided in the Generating Enterprise-wide Database User Directory topic.

The following columns are displayed on the report:

1. **Login Name –** Name of the database server login (SQL Server, ASE) or database user or group (DB2, Oracle, MySQL)
2. **Login Type –** Type of login, such as USER, SQL LOGIN or WINDOWS GROUP.
3. **Disabled –** State of the login indicated by a check box. If checked, the login is disabled and cannot be used for database server connections. This value is not applicable to DB2 and MySQL servers.
4. **Expired –** Account/password expiration state. If checked, the login is expired and the user must change his or her password before a successful database server connection can be established. This value is not applicable to DB2 and MySQL servers.
5. **Server Alias –** Database server alias specified in the DB Audit settings, typically the name of the database connection profile.
6. **Server Name –** The system server name as defined in the connection parameters. Additional values such as a port number can be added to the name to uniquely indentify the server.
7. **Server Type –** Type of database server, such as Oracle or MySQL.
8. **Snapshot Time – T**ime at which the security snapshot was generated.

Security Settings Comparison by Server

To run the report, select **Tools > Security Snapshot** menu and follow the instructions provided in the Auditing and Documenting Security Changes, Enforcing Change Control topic.

The report compares security and audit settings for two selected database servers using their most recent snapshots.

This report may contain several sections. The number of sections that appear in the report is data driven. For example, if changes have been made in database server logins or users, the Logins / Users section will be included in the report.

The following sections and columns may appear on the report:

Headers for Server 1 and Server 2

1. **Server Alias –** Database server alias specified in the DB Audit settings, typically the name of the database connection profile.
2. **Server Name –** System server name as defined in the connection parameters. Additional values such as a port number can be added to the name in order to uniquely indentify the server.
3. **Snapshot # -** A unique ID assigned to the security snapshot captured for this server and used for the comparison.

4. **Snapshot Time –** The time at which the security snapshot was generated.

Logins / Users

1. **Login Name –** Name of the database server login (SQL Server, ASE) or database user or group (DB2, Oracle, MySQL)
2. **Login Type –** Type of the login, such as USER, SQL LOGIN or WINDOWS GROUP.
3. **Change Type –** The type of the change, such as 'New login', 'dropped login' or 'login properties change'. Server 1 is considered as a base-line; settings for server 2 are compared against settings for server 1.
4. **Disabled 1 –** A checkbox indicating the state of the login on Server 1. If checked, the login is disabled and cannot be used for database server connections. This value is not applicable to DB2 and MySQL servers.
5. **Disabled 2 –** A checkbox indicating the state of the login on Server 2. If checked, the login is disabled and cannot be used for database server connections. This value is not applicable to DB2 and MySQL servers.
6. **Expired 1 –** Account/password expiration state of the login on Server 1. If checked, the login is expired and the user must change his or her password before a successful database server connection can be established. This value is not applicable to DB2 and MySQL servers.
7. **Expired 2 –** Account/password expiration state of the login on Server 2. If checked, the login is expired and the user must change his or her password before a successful database server connection can be established. This value is not applicable to DB2 and MySQL servers.

Security and System Audit Settings

1. **Parameter Category – A p**arameter category, such as 'SYSTEM CONFIG', 'AUDIT FILTER' or 'AUDIT CONFIG'. Parameter category names may differ for different types of database severs. The availability of specific categories depends on the physical organization of the database server security system and the system-level auditing.
2. **Parameter Name –** Name of the security parameter. Parameter names may differ for different types of database servers. The availability of specific parameters depends on the physical organization of the database server security system and the system-level auditing.
3. **Change Type –** Type of the change, such as 'New parameter,' 'dropped parameter,' or 'parameter value change'. Server 1 is considered to be the baseline; settings for server 2 are compared against settings for server 1.
4. **Value 1 –** Parameter value in Server 1/snapshot 1.
5. **Value 2 –** Parameter value in Server 2/snapshot 2.
6. **Parameter Description –** Description of the parameter.

Data-change Audit Settings

1. **Table Name –** Fully qualified name of the business table in the database.
2. **Change Type –** Type of the change, such as 'Table found in snapshot 1 only' , 'Table found in snapshot 2 only', 'parameter values don't match'. Server 1/snapshot 1 is considered to be the baseline; data-change audit settings for server 2/snapshot 2 are compared against settings for server 1.
3. **Audit Settings 1 –** Group of audit parameters for the table as they appear in Server 1/snapshot 1. Individual parameters are displayed within the group as separate lines in the format: *parameter*=*value*.
4. **Audit Settings 2 –** Group of audit parameters for the table as they appear in Server

2/snapshot 2.. Individual parameters are displayed within the group as separate lines in the format: *parameter*=*value*.

Data-change Audit Filters by Application

1. **Table Name –** Fully qualified name of the business table in the database.
2. **Change Type –** Type of the change, such as 'Filter found in snapshot 1 only' , 'Filter found in snapshot 2 only', 'parameter values don't match'. Server 1/snapshot 1 is considered to be the baseline; data-change audit settings for server 2/snapshot 2 are compared against settings for server 1.
3. **Audit Settings 1 –** Group of audit filter parameters for the table as they appear in Server 1/snapshot 1. Individual parameters are displayed within the group as separate lines in the format: *parameter*=*value*.
4. **Audit Settings 2 –** Group of audit filter parameters for the table as they appear in Server 2/snapshot 2. . Individual parameters are displayed within the group as separate lines in the format: *parameter*=*value*.

Data-change Audit Filters by Login/User

1. **Table Name –** Fully qualified name of the business table in the database.
2. **Change Type –** Type of the change, such as 'Filter found in snapshot 1 only', 'Filter found in snapshot 2 only', 'parameter values don't match'. Server 1/snapshot 1 is considered to be the baseline; data-change audit settings for server 2/snapshot 2 are compared against settings for server 1.
3. **Audit Settings 1 –** Group of audit filter parameters for the table as they appear in Server 1/snapshot 1. Individual parameters are displayed within the group as separate lines in parameter=value format.
4. **Audit Settings 2 –** Group of audit filter parameters for the table as they appear in Server 2/snapshot 2. . Individual parameters are displayed within the group as separate lines in the format: *parameter*=*value*.

Effective Security Settings

1. **Login Name –** Name of the database server login (SQL Server, ASE) or database user or group (DB2, Oracle, MySQL)
2. **Login Type –** Type of the login, such as USER, SQL LOGIN or WINDOWS GROUP.
3. **Change Type –** Type of the change, such as 'New login', 'dropped login', 'login properties change'. Server 1 is considered to be the baseline; settings for server 2 are compared against settings for server 1.
4. **Security Settings 1 –** Fully qualified user privilege in Server 1/snapshot 1. The value format depends on the effective privilege type and may differ for different types of privileges (depending on the privilege scope) and for different types of database systems.
5. **Security Settings 2 –** Fully qualified user privilege in Server 2/snapshot 2. The value format depends on the effective privilege type and may differ for different types of privileges (depending on the privilege scope) and for different types of database systems.

Security Settings Changes

To run the report, select **Tools > Security Snapshot** menu and follow the instructions provided in

the Auditing and Documenting Security Changes, Enforcing Change Control topic.

The report compares security and audit settings for two selected snapshots for the same database server.

This report may contain several sections. The number of sections that appear in the report is data driven. For example, if changes have been made in database server logins or users, the Logins / Users section will be included in the report. For description of the report sections and displayed columns see Security Settings Comparison by Server topic.

Enterprise-Wide Security Settings Changes

To run the report, select **Tools > Security Snapshot** menu and follow the instructions provided in the Auditing and Documenting Security Changes, Enforcing Change Control topic.

This report can be used to investigate and document security and audit settings changes made over time in all database servers registered for snapshot data collection. When running this report, you can specify the time period for the change analysis. For each server, DB Audit automatically finds the last available snapshot before the start of the period and the last available snapshot before the end of the period. It then performs a comparison of the two snapshots.

Comparison results for all servers are merged and the output included on the same report. The report output format is similar to the output format of the Security Settings Comparison by Server report and the Security Settings Changes report. For a description of the report sections and displayed columns, see Security Settings Comparison by Server topic.

Scheduled Audit Reports

The DB Audit Alert Center provides functions for scheduling automatic reports to be run at predetermined times and delivered by email. The Alert Center allows scheduling of most of the graphical reports available in the DB Audit Management Console. It also supports a separate group of aggregate reports that are only available through the Alert Center scheduling interface. In addition, the Alert Center supports running reports created in external reporting tools such as Crystal Reports.

Scheduling Reports

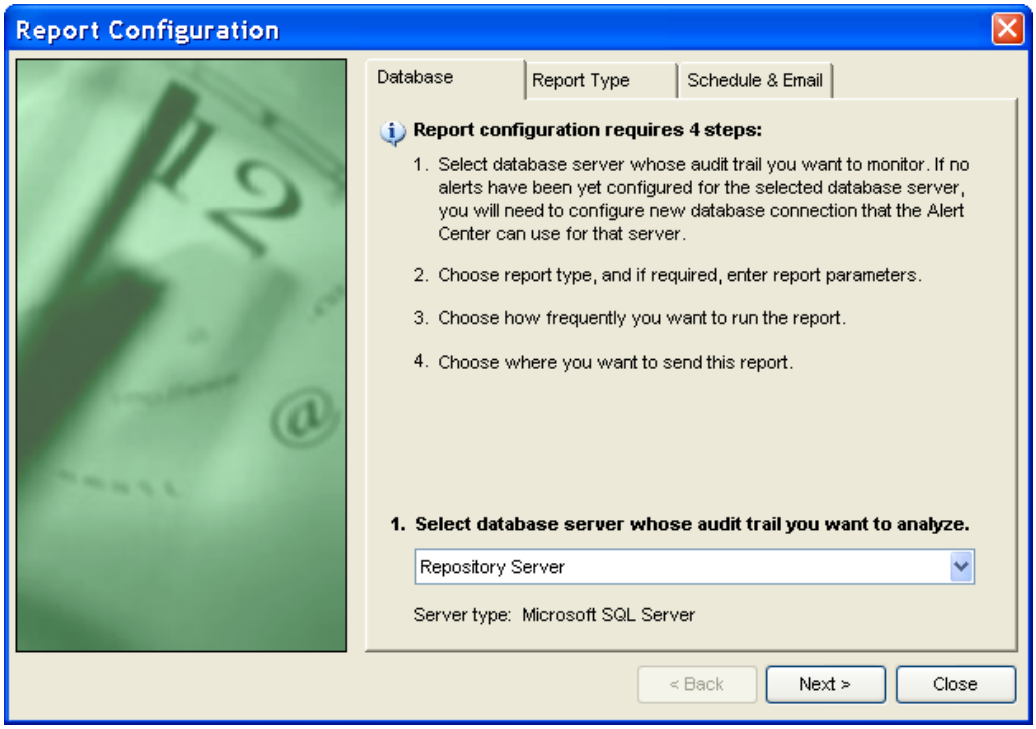
Use the Alert Center Remote Console interface to create and schedule automatic reports.

To create a new report:

1. Choose the data source for the report – In the Alert Center Remote Console interface, select either of the following items in the system tree:
    - Alerts and Reports from Central Repository Server
    - Alerts and Reports from Audited Database Server

2. Click the **Alerts > New Report** menu, or press **CTRL+K** hot key, or click the **New** button on the toolbar. The **Report Configuration** dialog will appears as shown in the example screenshot of the **Report Configuration** dialog).

3. The remaining steps required to schedule a new report are the same as steps required to setup a new alert. See Creating Alerts topic in CHAPTER 6 for more details..



Types of Reports That Can Be Scheduled

The following report types can be scheduled:

1. **Consolidated Events for the Previous Day** (fixed, single day reporting period)**:**
   Failed Connection Attempts (Yesterday)
   Successful Connections (Summary; Yesterday)
   Security Settings Changes (Yesterday)
   Audit Settings Changes (Yesterday)
   Database Schema Changes (DDL; Yesterday)
   Unauthorized Access Attempts (Yesterday)

2. **System & User Activities** (user-specified reporting period)**:**
   Logon/Logoff and Resource Usage Audit
   Object Access and Operations Audit
   Object Access Audit Summary
   Operations Audit Detail
   Operations Audit Summary
   User Activity (Failed Logons)
   User Activity (Last Logon Time)
   User Activity (Denied Access to Objects)

User Activity (Sys Admins)

3. **Compliance** (user-specified reporting period)**:**
   Recently Created, Deleted and Modified Users and Logins
   Recently Granted and Revoked Privileges
   Inactive Users with Active Accounts
   Users with Expired Passwords
   Users with Non-expiring Passwords
   Users Having Administrative Privileges
   Recent Administrator Logins
   Recent Privileged Operations (Create, Drop, Alter)

4. **Data-change Auditing** (user-specified reporting period)**:**
   Audit Trail Detail by Schema
   Audit Trail Detail by Application
   Audit Trail Detail by Table
   Audit Trail Summary By Table
   Audit Trail Summary for All Tables

5. **Other** (user-specified reporting period)**:**
   Database Errors
   Text of SQL Queries

6. **Crystal Reports and Other External Reports** (user-specified reporting period)**:**
   See Creating Custom Reports Using Crystal Reports topic for details

Run-time Report Controls

The Alert Center supports several parameters that allow you to control how scheduled reports are generated and sent. These parameters are stored in the **ac.prop** file located in the Alert Center installation directory. If this file doesn't exist, the Alert Center uses all defaults. ac.prop is a text file that can be created in Windows Notepad, the Unix vi editor or any other text editor. Users of DB Audit version 4.0 and later can also use the Alert Center graphical Remote Console utility to update these and other parameters remotely. Use the **File > Set Report Options** menu to view and modify the control parameters stored in the ac.prop file.

The ac.prop parameter values must be formatted as shown below:

PARAMETER=VALUE

No spaces are allowed before or after the equal sign. The following parameters are supported:

REPORT_MAX_ROWS – This parameter controls the maximum number of lines that can be displayed on the report. If the report query returns more then the specified number, the report is truncated after the specified number of lines, and a message at the bottom of the report indicates how many more lines were found but not displayed. The default value for this parameter is 1000 lines.

REPORT_ZIP_SIZE – This parameter controls the report email format. By default, all reports are sent as HTML email attachments. Values that can be specified for this parameter are:

0	Zip all reports regardless of size
-1	Do not zip reports (the default value)
n	File size in bytes. If the generated report exceeds the specified size, it is sent as a zipped file; otherwise, it is sent unzipped.

Example **ac.prop** file contents:

```
REPORT_MAX_ROWS=5000
REPORT_ZIP_SIZE=64000
```

Failed Connection Attempts (Yesterday)

This summary report analyzes data from the system audit trail and returns names of users/logins who were denied database access for any reason on the previous day.

While the report may indicate possible hacking attempts, it could also be a result of harmless attempts to enter an incorrect password. The Auditor should use personal judgment to decide which of the above situations occurred.

The following columns are displayed on the report:

1. **Server Name** – Name of the database server and instance where the failed attempt has been made. This value is only provided if the report is run from the central repository.
2. **OS User Name –** Network user name of the database user who has attempted to establish a new database connection.
3. **User Name –** Name of the database user who has attempted to establish a new database connection.
4. **Terminal –** Name of the network terminal or user workstation from which the failed attempt to make a new connection was made; for DB2, this is the network protocol-specific address of the network terminal or user workstation.
5. **Event Time –** The date and time when the failed attempt was made.

 **Notes:** This report requires that system-level auditing is installed and that auditing of failed logins is enabled.

The report scope depends on where the report is run. If the report is run in an audited database, it only covers events from that database. If the report is run in a central repository, it covers all database servers registered with the repository.

Successful Connections (Summary; Yesterday)

This summary report analyzes data from the system audit trail and returns names of users/logins who successfully gained access to the database on the previous day.

The report may include any or all of the users listed below. The Auditor should use personal judgment to decide which of the above situations occurred.

Authorized users who have valid access rights
Non-authorized users who are not supposed to have access to the database
Users who connect to the system after normal business hours
Users who share their workstations and terminals, letting other users run database programs on their systems

The following columns are displayed on the report:

1. **Server Name** – Name of the database server and instance where the successful connection was established. This value is only provided if the report is run from the central repository.

2. **OS User Name –** Network user name of the database user who established the successful database connection.
3. **User Name –** Name of the database user who established the successful database connection.
4. **Terminal –** Name of the network terminal or user workstation from which the successful database connection was established; for DB2, the network protocol-specific address of the network terminal or user workstation.
5. **Total Connects** – The number of successfully established database connections for the user/terminal pair of values during the reporting period.
6. **First Event Time –** The date and time when the first database connection has been established for the user/terminal pair of values during the reporting period.
7. **Last Event Time –** The date and time when the first database connection was established for the user/terminal pair of values during the reporting period. The First and Last Event Times show the time period over which all successful connections have been established yesterday.

 **Notes:** This report requires that the system-level auditing is installed and that auditing of successful logins is enabled.

The report scope depends on where the report is run. If the report is run in an audited database, it only covers events from that database. If the report is run in a central repository, it covers all database servers registered with the repository.

Security Changes (Yesterday)

This summary report analyzes data from the system audit trail and returns all attempts to modify database security settings and accounts that occurred on the previous day.

The report may show both authorized operations and also non-authorized operations. The report may also show users with elevated privileges who are not supposed to have them as well as various kinds of hacking attempts. The Auditor should use personal judgment to decide which of the above situations occurred.

The report output differs for different database systems. The following columns are displayed on the report:

 **Central Repository based-reporting:**

1. **Server Name** – Name of the database server and instance where the change was attempted.
2. **OS User Name –** Network user name of the database user who attempted to make security changes.
3. **User Name –** Name of the database user who attempted to make security changes.
4. **Terminal –** Name of the network terminal or user workstation from which the attempt was made; for DB2, the network protocol specific address of the network terminal or user workstation.
5. **Event Time –** The date and time when the change was made.
6. **Action Name –** The type of operation performed such as ALTER, CREATE, or DROP user or logon; GRANT/DENY/REVOKE access, role or privilege; ALTER logon properties and so on.
7. **Object Type –** If changes have been made to schema object scope, this value indicates the type of the object whose security settings have been altered. Object Type is not available for reporting in all supported database systems.

8. **Schema –** If changes have been made to schema object scope, this value indicates the schema of the object whose security settings have been altered.
9. **Object Name –** If changes have been made to schema object scope, this value indicates the name of the object whose security settings have been altered.
10. **App Name –** Name of the application that was used to make security changes. App Name is available only in audit records generated in SQL Server database systems.

 **SQL Server**:

1. **Login Name –** Login user name of the database user who attempted to make security changes.
2. **DB Name –** Name of the database where the security change was made. Note that "master" database may also indicate global server-wide security changes such as creation or alteration of a login.
3. **Action Name –** The type of operation performed such as ALTER, CREATE, or DROP user or logon; GRANT/DENY/REVOKE access, role or privilege; ALTER logon properties and so on.
4. **Event Time –** The date and time when the change was made.
5. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
6. **OS User Name –** Network user name of the database user who attempted to make security changes.
7. **App Name –** Name of the application that was used to make security changes. App Name is available only in audit records generated in SQL Server database systems.
8. **Target –** Target of the security change; for example, login name or object name.
9. **SQL Command –** The SQL command that was used to make the changes.
10. **Success –** Indicates whether the change has been successful or not.

 **Oracle**:

1. **User Name –** Name of the database user who attempted to make security changes.
2. **Action Name –** The type of operation performed such as ALTER, CREATE, or DROP user or logon; GRANT/DENY/REVOKE access, role or privilege; and so on.
3. **Event Time –** The date and time when the change has been made.
4. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
5. **OS User Name –** Network user name of the database user who attempted to make security changes.
6. **Target –** Target of the security change; for example, user name or object name.
7. **Grantee –** Name of the database user who is the grantee of the security changes.
8. **Priv Used –** The effective database privileges that grantor used to make the changes.
9. **Status –** Indicates whether the change was successful or not.

 **ASE**:

1. **Login Name –** Login user name of the database user who attempted to make security changes.

2. **Event Time –** The date and time when the change was made.
3. **Action Name –** The type of operation performed such as ALTER, CREATE, or DROP user or logon; GRANT/DENY/REVOKE access, role or privilege; ALTER logon properties; and so on.
4. **Additional Info –** Additional information describing privileges used by the user to make the changes.
5. **DB Name –** Name of the database where the security change was made. Note that "master" database may also indicate global server-wide security changes such as creation or alteration of a login.
6. **Schema –** If changes have been made to schema object scope, this value indicates the schema of the object whose security settings have been altered.
7. **Object Name –** If changes have been made schema object scope, this value indicates name of the object whose security settings have been altered.

 **DB2**:

1. **User Name –** Name of the database user who attempted to make security changes.
2. **DB Name –** Name of the database where the security change was made.
3. **Object Type –** If changes have been made to schema object scope, this value indicates type of the object whose security settings have been altered.
4. **Schema –** If changes have been made to schema object scope, this value indicates the schema of the object whose security settings have been altered.
5. **Object Name –** If changes have been made to schema object scope, this value indicates the name of the object whose security settings have been altered.
6. **Action Name –** The type of operation performed such as ALTER, CREATE, or DROP user or logon; GRANT/DENY/REVOKE access, role or privilege; and so on.
7. **Event Time –** The date and time when the change was made.
8. **Terminal –** Network protocol specific address of the network terminal or user workstation from which the attempt was made.
9. **OS User Name –** Network user name of the database user who attempted to make security changes.
10. **Privilege Name –** This indicates the privileges that were used to make changes.
11. **App Name –** Identifier of the application that was used to make security changes.
12. **Grantee –** Name of the database user who is the grantee of the security changes.
13. **Status –** This indicates whether the change was successful or not.

 **Notes for all systems:** This report requires that system-level auditing is installed and that auditing of all security related operations is enabled. Alternatively, in SQL Server you can enable auditing of text of executed SQL queries. The report will then search the system audit trail for predetermined security events as well as scan recorded SQL queries for known security-related commands.

The report scope depends on where the report is run. If the report is run in an audited database, it only covers events from that database. If the report is run in a central repository, it covers all database servers registered with the repository.

Audit Settings Changes (Yesterday)

This summary report analyzes data from the system audit trail and returns all attempts to modify audit settings that occurred on the previous day.

The report may show both authorized operations and non-authorized operations. The Auditor should use personal judgment to decide which of the above situations occurred.

The report output differs for different database systems. The following columns are displayed on the report:

 **Central Repository based-reporting:**

1. **Server Name** – Name of the database server and instance where the change has been attempted.
2. **OS User Name –** Network user name of the database user who attempted to make audit settings changes.
3. **User Name –** Name of the database user who attempted to make audit settings changes.
4. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
5. **Event Time –** The date and time when the change was made.
6. **Action Name –** The type of operation performed such as AUDIT, NOAUDIT, ALTER, CREATE, or DROP audit-related object or privilege and so on.
7. **Object Type –** In the case of an object-level change, this value indicates the type of object affected by the change. Object Type is not available for reporting in all supported database systems.
8. **Schema –** In the case of an object-level change, this value indicates the schema of the object affected by the change.
9. **Object Name –** In the case of an object-level change, this value indicates the name of object affected by the change.
10. **App Name –** Name of the application that was used to make audit changes. App Name is available only in audit records generated in SQL Server database systems.

 **SQL Server**:

1. **Login Name –** Login user name of the database user who attempted to make audit changes.
2. **DB Name –** Name of the database where the change was made.
3. **Action Name –**The type of operation performed such as DELETE, UPDATE, CREATE, ALTER, DROP and so on.
4. **Event Time –** The date and time when the change was made.
5. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
6. **OS User Name –** Network user name of the database user who attempted to make audit changes.
7. **App Name –** Name of the application that was used to make audit changes.
8. **Target –** Target of the audit change such as login name or object name.
9. **SQL Command –** The SQL command that has been used to make the changes.
10. **Success –** Indicates whether the change was successful or not.

Oracle:

1. **OS User Name –** Network user name of the database user who attempted to make audit changes.
2. **User Name –** Name of the database user who attempted to make audit changes.
3. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
4. **Event Time –** The date and time when the change was made.
5. **Action Name –** The type of operation performed such as AUDIT, NOAUDIT, CREATE, ALTER, DROP and so on.
6. **Object Name –** In the case of an object-level change, this value indicates name of object affected by the change.

ASE, DB2: This report is not supported by ASE or DB2.

Notes for all systems: This report requires that system-level auditing be installed and that auditing of all DDL and security-related operations be enabled. Alternatively, in SQL Server you can enable auditing of text of executed SQL queries. The report will then search the system audit trail for predetermined events and will scan recorded SQL queries for known commands.

The report scope depends on where the report is run. If the report is run in an audited database, it only covers events from that database. If the report is run in a central repository, it covers all database servers registered with the repository.

Database Changes (DDL; Yesterday)

This summary report analyzes data from the system audit trail and returns all structural database changes (so called DDL - Data Definition Changes) made on the previous day.

The report may show both authorized operations and also non-authorized operations. The Auditor should use personal judgment to decide which of the above situations occurred.

The report output differs for different database systems. The following columns are displayed on the report:

Central Repository based-reporting:

1. **Server Name** – Name of the database server and instance where the changes were made.
2. **OS User Name –** Network user name of the database user who made the changes.
3. **User Name –** Name of the database user who attempted to make changes.
4. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
5. **Event Time –** The date and time when the change was made.
6. **Action Name –** The type of operation performed such as ALTER, CREATE, DROP and so on.
7. **Object Type –** Type of object affected by the change. Object Type is not available for reporting in all supported database systems.

8. **Schema –** If applicable, schema of the object affected by the change.
9. **Object Name –** Name of object affected by the change.
10. **App Name –** Name of the application that was used to make changes. App Name is available only in audit records generated in SQL Server database systems.

 **SQL Server**:

1. **OS User Name –** Network user name of the database user who attempted to make changes.
2. **Login Name –** Login user name of the database user who attempted to make changes.
3. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
4. **Event Time –** The date and time when the change was made.
5. **DB Name –** Name of the database where the changes were made.
6. **Action Name –** The type of operation performed, such as ALTER, CREATE, DROP and so on.
11. **Object Type –** Type of the object affected by the change. Object Type is not available for reporting in all supported database systems.
12. **Schema –** If applicable, schema of the object affected by the change.
13. **Object Name –** Name of object affected by the change.
7. **App Name –** Name of the application that was used to make security changes.

 **Oracle**:

1. **OS User Name –** Network user name of the database user who attempted to make changes.
2. **User Name –** Name of the database user who attempted to make audit changes.
3. **Terminal –** Name of the network terminal or user workstation from which the attempt was made.
4. **Event Time –** The date and time when the change was made.
5. **Action Name –** The type of operation performed such as ALTER, CREATE, DROP and so on.
6. **Object Name –** In the case of an object-level change, this value indicates the name of the object affected by the change.

 **ASE**:

1. **Login Name –** Login user name of the database user who attempted to make DDL changes.
2. **Event Time –** The date and time when the change was made.
3. **Action Name –** The type of operation performed such as ALTER, CREATE, DROP and so on.
4. **Additional Info –** Additional information describing privileges used by the user to make the changes.
5. **DB Name –** Name of the database where the change was made.
6. **Schema –** If applicable, schema of the object affected by the change.

7. **Object Name –** In the case of an object-level change, this value indicates the name of the object affected by the change.

DB2:

1. **User Name –** Name of the database user who attempted to make DDL changes.
2. **DB Name –** Name of the database where the change was made.
3. **Schema –** If changes have been made to schema object scope, this value indicates the schema of the object whose security settings have been altered.
4. **Object Name –** If changes have been made to schema object scope, this value indicates the name of the object whose security settings have been altered.
5. **Action Name –** The type of operation performed such as ALTER, CREATE, DROP and so on.
6. **Event Time –** The date and time when the change was made.
7. **Terminal –** Network protocol specific address of the network terminal or user workstation from which the attempt was made.
8. **OS User Name –** Network user name of the database user who attempted to make changes.
9. **App Name –** Identifier of the application that was used to make changes.

Notes for all systems: This report requires that system-level auditing is installed and auditing of all DDL operations is enabled.

The report scope depends on where the report is run. If the report is run in an audited database, it only covers events from that database. If the report is run in a central repository, it covers all database servers registered with the repository.

Unauthorized Access Attempts (Yesterday)

This summary report analyzes data from the system audit trail and returns all failed attempts to access database objects that occurred on the previous day.

The report output differs for different database systems. The following columns are displayed on the report:

Central Repository based-reporting:

1. **Server Name** – Name of the database server and instance where the change was attempted.
2. **OS User Name –** Network user name of the database user who attempted to make changes.
3. **User Name –** Name of the database user who attempted to make changes.
4. **Terminal –** Name of the network terminal or user workstation from which the attempt was made; for DB2, the network protocol-specific address of the network terminal or user workstation.
5. **Event Time –** The date and time when the change was attempted.
6. **Action Name –** The type of operation attempted such as ALTER, CREATE, or DROP

user or logon; GRANT/DENY/REVOKE access, role or privilege; ALTER logon properties; DML operations such as SELECT, UPDATE, DELETE; and so on.

7. **Object Type –** If a change was attempted for a schema object, this value indicates the type of the object whose settings or data have been affected. Object Type is not available for reporting in all supported database systems.
8. **Schema –** If changes have been made to schema object scope, this value indicates the schema of the object whose settings or data have been affected.
9. **Object Name –** If changes have been made to schema object scope, this value indicates name of the object whose settings or data have been altered.
10. **App Name –** Name of the application that was used to make changes. App Name is available only in audit records generated in SQL Server database systems.

 **SQL Server, ASE**:

1. **Event Time –** The date and time when the change was attempted.
2. **OS User Name –** Network user name of the database user who attempted to make changes.
3. **Login Name –** Login user name of the database user who attempted to make changes.
4. **Object Name –** If applicable, full name of the database object affected.
5. **Action Name –** The type of operation attempted such as ALTER, CREATE, or DROP user or logon; GRANT/DENY/REVOKE access, role or privilege; ALTER logon properties; DML operations such as SELECT, UPDATE, DELETE; and so on.

 **Oracle, DB2**:

1. **Event Time –** The date and time when the change was attempted.
2. **OS User Name –** Network user name of the database user who attempted to make changes.
3. **User Name –** Name of the database user who attempted to make security changes.
4. **Object Name –** If applicable, full name of the database object affected.
5. **Action Name –** The type of operation attempted such as ALTER, CREATE, or DROP user or logon; GRANT/DENY/REVOKE access, role or privilege; ALTER logon properties; DML operations such as SELECT, UPDATE, DELETE; and so on.

 **Notes for all systems:** This report requires that system-level auditing is installed and that auditing of all security related operations is enabled. Alternatively, in SQL Server you can enable auditing of text of executed SQL queries. The report will then search the system audit trail for predetermined security events and will scan recorded SQL queries for known security-related commands.

The report scope depends on where the report is run. If the report is run in an audited database, it only covers events from that database. If the report is run in a central repository, it covers all database servers registered with the repository.

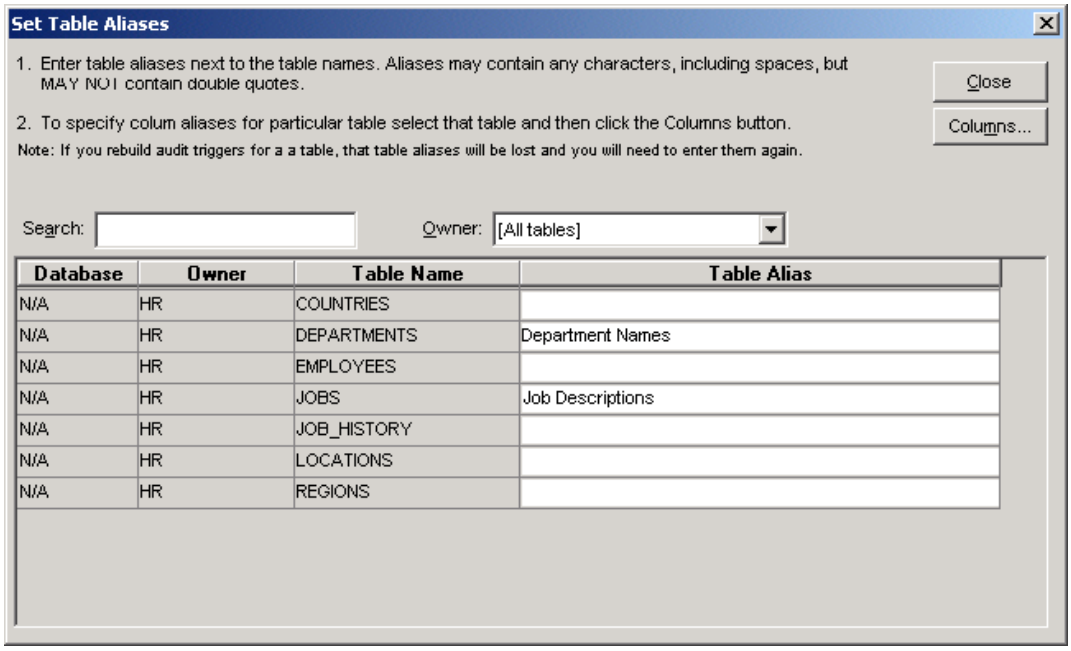
Working With Interactive Reports

Setting Table and Column Aliases

DB Audit allows you to define table and column aliases you can use in data-change audit reports in place of the non-descriptive table and column names commonly used in business tables. Using this feature, you can provide your users with flexible, user-friendly reports that do not require them to know the physical database design. For details on how to create custom reports, refer to the Custom Reports topic.

To setup table aliases

1. Select the **Data Audit > Set Table and Column Aliases** menu, or click the **Aliases** button on the Toolbar. The **Set Table Aliases** dialog will appear.



2. Enter table aliases.

3. To enter column aliases for a particular table, click the table name, then click the **Columns** button. The **Set Column Aliases** dialog will appear. Enter column aliases and press the **Close** button to close the **Set Column Aliases** dialog.

4. If needed repeat step 3 for other tables.

5. Click the **Close** button when done.

Searching Report Data

To search report results:

1 Select **View > Find Text** menu or click the **Find** button on the Toolbar. The Search dialog box will appear.

2 Type in the text you want to search for.

3 Choose the desired search options. See the following paragraphs for a description of the available search options.

4 Click the **Find Next** button to start searching. Note that search is performed in all columns of the active report. The search is case insensitive.

5 If the specified text is found, the search dialog box stays on top so you can review the search results and continue searching if needed. Text that is found is highlighted on the report, and the report is automatically advanced to the page where the text is found. Keep clicking the **Find Next** button to continue searching for subsequent occurrences of the specified text; otherwise click the **Close** button to stop searching.

Search options

Search Backwards – Searches backwards starting from the last report row selected or from the row containing the last occurrence of found text. If no row is selected by either of these means, the search proceeds from the last report row.

Wrap at Beginning or End – The search is performed for the entire report. When the search reaches either the end or the beginning of the report (depending on the search direction), the search continues from the opposite extreme until it returns to the row where the search was initiated.

Use As Pattern - DB Audit uses the character string entered in the Search box as a search pattern. A search pattern consists of **metacharacters**, which have special meaning in the match string, and **non-metacharacters**, which match the characters themselves. The following tables explain the meaning and use of the metacharacters:

Metacharacter	Meaning	Example
Caret (^)	Matches the beginning of a string	^C matches C at the beginning of a string
Dollar sign ($)	Matches the end of a string	s$ matches s at the end of a string
Period (.)	Matches any character	. . . matches any three consecutive characters
Backslash (\)	The escape character; indicates that the following metacharacter should be treated as a non-metacharacter	\$ matches $
Character class (a group of characters enclosed in square brackets ([ ])	Matches any of the enclosed characters	[AEIOU] matches the uppercase characters A, E, I, O, or U. You can use a hyphen to abbreviate a range of characters within a character class. For example, [A-Za-z] matches any upper- or lowercase alphabetic character

Complemented character class (first character inside the brackets is a caret)	Matches any character not in the group following the caret	[^0-9] matches any character except a digit. [^A-Za-z] matches any character except an alphabetic character

The metacharacters asterisk (*), plus (+), and question mark (?) are unary operators used to indicate repetition in regular expressions:

Metacharacter	Meaning	Example
* (asterisk)	Indicates zero or more occurrences	A* matches zero or more As (no A, A, AA, AAA, and so on)
+ (plus)	Indicates one or more occurrences	A+ matches one A or more than one A (A, AAA, and so on)
? (question mark)	Indicates zero or one occurrence	A? matches an empty string ("") or A

Example patterns

The following table shows various text patterns and example text that matches each pattern:

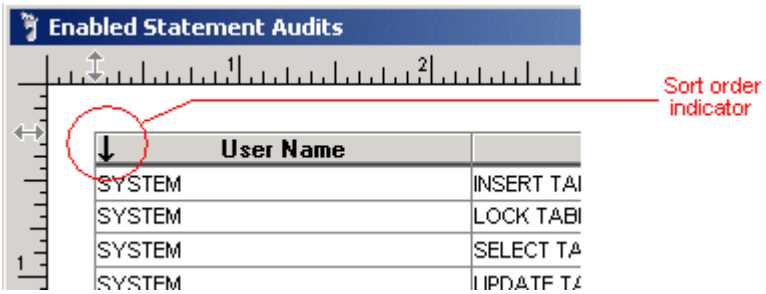
This pattern	Matches
AB	Any string that contains AB; for example, ABA, DEABC, textAB_one
B*	Any string that contains zero or more Bs; for example, AC, B, BB, BBB, ABBBC, and so on
AB*C	Any string containing the pattern AC or ABC or ABBC, and so on (zero or more Bs)
AB+C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (one or more Bs)
ABB*C	Any string containing the pattern ABC or ABBC or ABBBC, and so on (one B plus zero or more Bs)
^AB	Any string starting with AB
AB?C	Any string containing the pattern AC or ABC (zero or one B)
^[ABC]	Any string starting with A, B, or C
[^ABC]	A string containing any characters other than A, B, or C
^[^abc]	A string that begins with any character except a, b, or c
^[^a-z]$	Any single-character string that is not a lowercase letter (^ and $ indicate the beginning and end of the string)
[A-Z]+	Any string with one or more uppercase letters
^[0-9]+$	Any string consisting only of digits
^[0-9][0-9][0-9]$	Any string consisting of exactly three digits

^([0-9][0-9][0-9])$	Any consisting of exactly three digits enclosed in parentheses

Sorting Reports

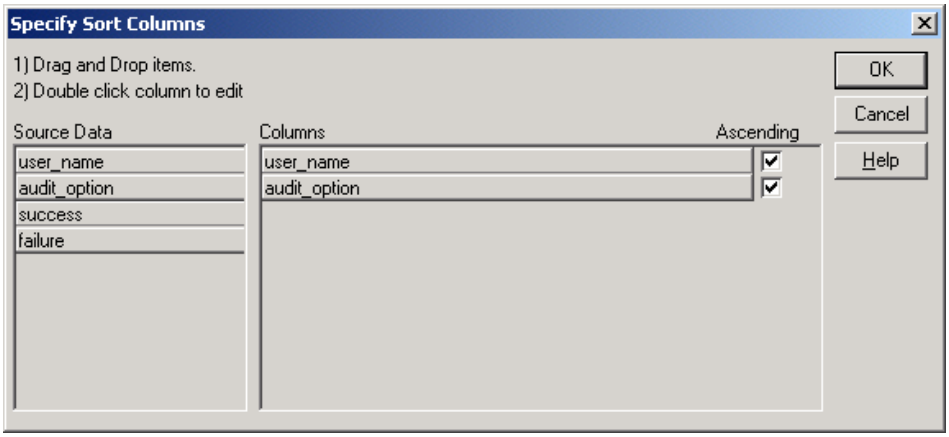
You can sort the rows of any report by the data in one or more adjacent columns.

For a simple sort by one column, click the header of the column you want to sort by. The down arrow appears over the header indicating that rows are sorted in ascending order. Click the header again to sort descending. The up arrow indicates that rowss are sorted in descending order.



For complex sorts by two or more columns or by regular expressions:Click the **View > Sort** menu or the **Sort** button on the Toolbar. The **Sort** dialog box will appear.

1.



2. Drag and drop columns from the source data list box to the columns list box.

3. Check the **Ascending** check box to sort the preceding column/expression in ascending order; or leave it blank to sort descending.

4. For an advanced sort, double-click the column/expression in the column list. The **Modify Expression** dialog box will appear. Enter the expression you want to use as as a search pattern, then press the **OK** button to apply the new sort. Press the **Verify** button to verify your expression syntax.

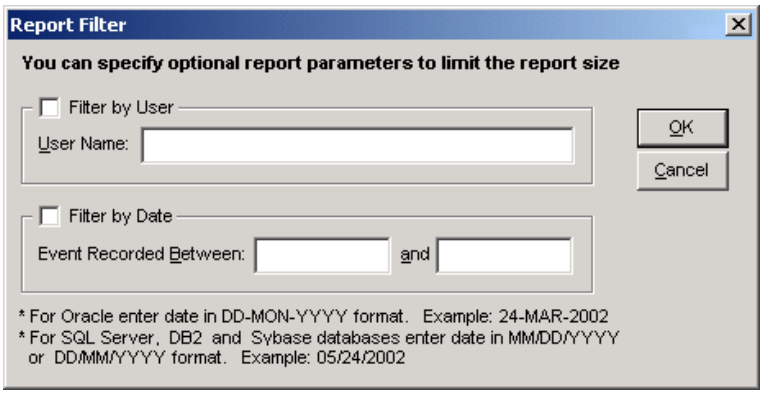
Filtering Reports

Filtering is used to limit the scope of audit data selected for inclusion in a DB Audit report. DB Audit supports two filtering methods:

- **Back-end filtering** – The filter is applied by the DBMS at the time the report data is generated. Back-end filtering is limited to three options that can be used simultaneously: filter by audited table, filter by user/login, and filter by date.
- **Front-end filtering** – The filter is applied by DB Audit after report data is extracted. This filtering method provides virtually unlimited possibilities for selecting report data, but keep in mind that filtering is performed in the local memory of the client computer used to preview the report. A front-end filter can be removed or modified as many times as needed without re-running the report.

Back-end filtering

Back-end filtering can consist of one or two steps. When the report is based on data from a data-change audit trail, you have the option to select a particular audited table to use for the report. You can then further limit the report by specifying the name of a user who you want to check and a range of dates when data changes might have been done. You can use the Report Filter dialog to specify optional user name and dates.

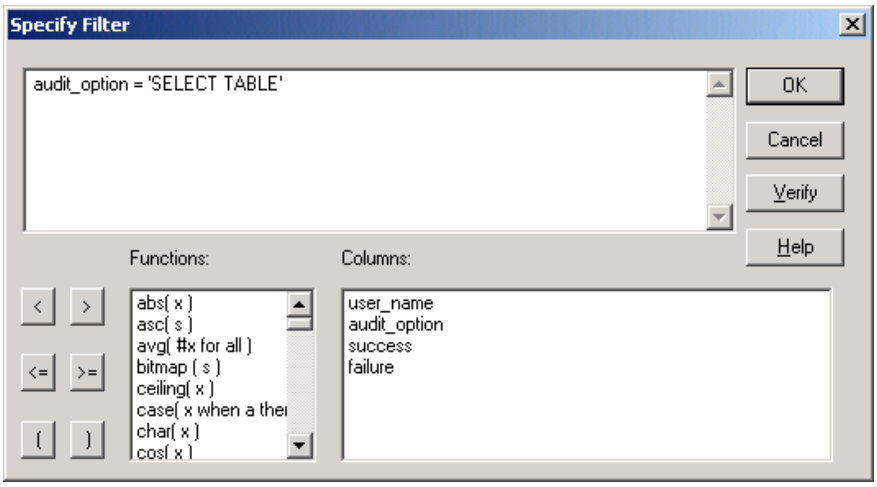


Notes:

- User and login names are case sensitive. In Oracle and DB2, enter names in upper case. In SQL Server, ASE and ASA enter names in the appropriate case. Most often login names in these databases are created in lower case.
- If you choose to specify event dates for the filter, make sure to enter dates in a valid format that is understood by your database. Failure to specify valid dates will lead to report error and cancellation.

Front-end filtering

To apply an advanced front-end filter after the report is retrieved, click the **View > Filter** menu or click the **Filter** button on the Toolbar.



Enter the filter expression using available report column names and supported functions. Press the **Verify** button to verify your filter expression. Press the **OK** button to apply the filter.

Example filter expressions:

```
audit_option = 'SELECT TABLE'
audit_option IN ('DELETE', 'UPDATE')
audit_option = 'SELECT' and user_name LIKE 'SAM%'
```

Exporting Report Data

Any DB Audit report can be exported to a number of popular file formats. The following export formats are supported:

File Extension	File Format
XLS	Microsoft Excel format
CSV	Comma-separated values
TXT	Tab-separated columns with a carriage return at the end of each row
XML	XML table with a default style sheet (XLS) transformation applied.
HTM	Text with HTML formatting (HTML table)
SQL	SQL syntax (SQL INSERT statements)

To export report data

1. Click **File > Save** menu or the **Save** button on the Toolbar. The standard **Select Export File Name** dialog will appear.
2. Select the desired export file type in the **File Type** drop-down box

3. Enter the name of the export file and press the **OK** button.

Zooming Reports In and Out

You can "zoom in" to get a close-up view of your data or "zoom out" to see more of the page at a reduced size. Click **View > Zoom** menu. A zoom dialog box will appear. Select the desired magnification and then click the **OK** button to apply the new zoom.

Printing Reports

Choosing a printer

Select **File > Print** Setup command. A Printer Setup dialog box will appear. From the displayed list of connected printers, select the desired printer and then click the **OK** button.

To prevent a document from flowing onto an additional printed page

If the report page is too wide to fit on one printed page and only a small amount of text appears on the next page, you may narrow column width for some columns.

To print a range of pages or print only specific pages

Under **Page Range**, specify the portion of the report you want to print. Enter page numbers or page ranges you want to include, or both. Use commas to separate individual pages and page ranges. For example, to print pages 1, 2, and 5 through 10, enter: "1,2, 5-10". An empty string means ”Print All”. From the **Range Include** drop-down list, select the pages to print within the desired range. Values are: ”Print All,” “Print All Even Pages,” “Print All Odd Pages.”

To print more than one copy at a time

Collate - This option controls whether the printing is collated. Note that collating is usually slower since the print is repeated to produce collated sets.
Copies - The number of copies to be printed.

Print Margins

Bottom - Width of the bottom margin on the printed page.
Left - Width of the left margin on the printed page.
Right - Width of the right margin on the printed page.
Top - Width of the top margin on the printed page.

Margin Marks

While previewing a report in DB Audit workspace, you can use the mouse to visually set print margins. Move the mouse pointer over the resize arrows displayed on the rulers. Press the left mouse key and hold it down while dragging the margin mark as needed to achieve the desired margin width.

Other Print Options

Orientation - Print orientation.
Paper Size - Size of the paper that will be used for the output.
Paper Source - The printer bin tto be used as the paper source.
Print Quality - Print output quality. Note that higher quality is usually slower.
Scale - An integer specifying the scale of the printed output as a percentage of the original. If you have problems with scaling, you may be using a driver that does not support scaling.

Custom Reports

Custom Report Types

DB Audit supports several facilities for creating custom reports. Using these facilities, you can:

- Create interactive graphical reports for data-change auditing. You can use the DB Audit Management Console to create, modify and run such reports. This method is most appropriate for end users who may not be familiar with the internal database designs or who do not have direct access to the database.

  Read User-Defined Report Library topic for detailed instructions on how to create and work with such reports.

- Create user-defined audit views in the repository database using any available database management or programming tools and use DB Audit Management Console to register such views as custom report data sources. You can then use other DB Audit Management Console to display and work with data-change audit reports populated from custom data sources. This method is most appropriate for database administrators who are tasked with audit reports setup and management.

  Read Creating Custom Reports Using Database Views topic for detailed instructions on how to create and work with such reports.

- Create non-interactive scheduled reports for both system auditing and data-change auditing and have them periodically delivered to your email Inbox. You can use the Alert Center Remote Console to create, modify and test such reports. This method can be used by any user as it does not require special knowledge of the database or audit setup and internal working.

  Read "Alert Center Remote Console" topic in CHAPTER 6 for detailed instructions on how to create and work with such reports.

- Schedule reports and analyze audit trail data using third-party commercial reporting tools such as Crystal Reports. This method can be used by database administrators and developers familiar with the database working.

  Read "Alert Center Remote Console" topic in CHAPTER 6 for detailed instructions on how to create and work with such reports.

- Access and analyze audit trail data using reporting tools that come readily with your database systems, including Oracle OLAP, Microsoft SQL Server Reporting Services, Sybase InfoMaker, and many others. This method can be used by database administrators and developers familiar with the database. This method provides maximum flexibility and features.

  Read "Creating a Data-change Audit Report in Microsoft Excel" topic and "Creating a System Audit Report in Microsoft Excel" topic in this chapter for instructions on how to create data-change and system audit trail reports in Microsoft Excel.

Creating Custom Reports Using Crystal Reports

Important Notes: Crystal Reports software is not included with DB Audit packages. You must obtain a separate license for Crystal Reports that is completely independent from DB Audit license. A Crystal Reports license can be obtained from SAP Business Objects Corporation, makers of the Crystal Reports or from their resellers.

To be able to run Crystal Reports unattended and have reports delivered by email using DB Audit Alert Center scheduling and report running facilities, you must also obtain a license for the Crystal Command utility, which is also available from Business Objects and their resellers. This utility must be installed on the Alert Center computer.

The following brief tutorial describes how to use Crystal Reports to analyze and report on the data in the audit trail.

1. On the Alert Center computer, configure an ODBC profile pointing to the repository database containing the audit trail data.

2. Launch the Crystal Reports Designer tool

3. Select as the data source the ODBC profile created in step 1.

4. Choose the audit trail tables you want to use in the report.

5. Drag the required table columns to the Report Design workspace and use the **Property** window or right-click menu to modify the object properties.

6. Add additional objects to the report as necessary. These objects include labels for the report header, labels for column names, formulas, group name fields, summary fields, charts and so on

7. Arrange objects so they appear in the correct report sections. For example, if you place a chart object in the Report Header section, the chart will appear only once at the beginning of the report and will summarize the data contained in the report. Alternatively, if you place a chart object into the Group Header or Footer section, a separate chart will appear at the beginning or ending of each group of data and will summarize the data relating only to that group.

8. Run the report to test that it is working well and that it returns the expected results.

9. Save your report as a file with the .RPT extension and copy it to the Alert Center computer.

10. Launch the Alert Center Remote Console and connect to the Alert Center.

11. Select any database system in the system tree and click the **Alerts > New Report** menu or click the **New Report** button on the Alert Center Remote Console toolbar. The Alert Center Remote Console will display **Report Configuration** dialog.

12. Click the **Next** button and choose **Custom Report Using Crystal Reports** item in the report type drop-down list.

13. Enter the report name and the command line. For the command line enter the following:

    ```
    C:\CML\CML.exe  C:\ReportFolder\MyReportName.rpt /X 31 /D 9 /ETO
    ToEmailAddress /ESB Report Name /EMS Report message.
    ```

    In this command replace the following:
    C:\CML -> replace with the path to the Crystal Reports Command Line utility
    C:\ReportFolder\MyReportName.rpt -> replace with the full name of the file copied in step 9
    ToEmailAddress - > replace with the report recipient email address
    Report Name -> replace with the report name. It is recommended that you use the same name you used for the Report Name property. This name will be used as the email message subject.
    Report message -> replace with whatever text you want to use for the report message. The length of the entire command line may not exceed 255 characters.

14. Click the **Next** button. Enter report schedule type and frequency. Enter the email recipient address again.

15. Click the **Finish** button.

Creating Custom Reports Using Database Views

Database administrators and developers can use this method to replace default data-audit trail reports generated by DB Audit (and pulling data directly from audit trail tables) with reports pulling data from custom user-defined audit views.

It is important to note that user-defined views can return a subset of columns from audit trail tables, add additional columns as required, job audit trail tables with other tables and views in the database, and basically use any valid SQL functions and expressions to format the data display as required by your internal company rules.

The following four requirements apply to custom audit views:

- Custom audit views must be created in the **DB_AUDIT** schema in the local audit repository database.

- In SQL Server, Sybase ASE and Sybase ASA databases, custom views must return at least two columns with the names **audit_timestamp** and **audit_login.** In Oracle, DB2 and MySQL, custom views must return **audit_timestamp** and **audit_user** columns. These columns are required by the DB Audit Management Console if an end user chooses to apply time or user-name report filters. Failure to add these columns to the custom audit view may lead to run-time report generation errors.

- Custom audit views must have names in the following format: **V_TableSchema_TableName**. Here the table schema and table name are schema and name of the table being audited. For example, if the full name of the audited table is HR.EMPLOYEE, the view name must be V_HR_EMPLOYEE and, as explained in the first requirement above, it must be created in the DB_AUDIT schema. The full view name, then, is DB_AUDIT.V_HR_EMPLOYEE.

- To get DB Audit Management Console, use the custom view as data-change audit report data-source, the view name must be entered as a table alias in DB Audit settings.

Example: Creating a custom report using a user-defined audit view

This example demonstrates how to create a view for audit trail for the HR.EMPLOYEE table, replace value for department ID column (DEPT_ID) with the department name in the report results, replace value for the manager ID column (MGR_ID) with the manager name, and finally register this view with DB Audit as a custom data source for audit trail report for HR.EMPLOYEE table. For the purpose of this example, the following simplified EMPLOYEE table structure is assumed.

```
CREATE TABLE HR.EMPLOYEE
(
    EMP_ID int,
    FIRST_NAME varchar(30),
    LAST_NAME varchar(30),
    HIRE_DATE datetime,
    DEPT_ID int,
    MGR_ID int
)
```

 **SQL Server, ASE:** This table is assumed to be in the BUSINESS database in a repository database with the name AUDIT. Before you create the audit view, make sure to set the current database context to the AUDIT database.

It is also assumed that the name of audit trail table for HR.EMPLOYEE IS DB_AUDIT.AUDIT_200709292308535150.

The following steps are required to to create a view for the HR.EMPLOYEE table.

1. Using any available database tool, connect to the database and set the context to the local audit repository database.

2. Create the following view

```
CREATE VIEW DB_AUDIT.V_HR_EMPLOYEE AS
SELECT a.AUDIT_TIMESTAMP,
       a.AUDIT_APPNAME,
       a.AUDIT_TERMINAL,
       a.AUDIT_LOGIN,
       a.AUDIT_STATEMENT,
       a.AUDIT_VALUE_TYPE,
       a.EMP_ID,
       a.FIRST_NAME,
       a.LAST_NAME,
       a.HIRE_DATE,
       d.DEPT_NAME,
       e.FIRST_NAME + ' ' + e.LAST_NAME as MGR_NAME
FROM DB_AUDIT.AUDIT_200709292308535150 a
    LEFT OUTER JOIN BUSINESS.HR.DEPT d
        ON d.DEPT_ID = a.DEPT_ID
    LEFT OUTER JOIN BUSINESS.HR.EMPLOYEE e
        ON e.EMP_ID = a.MGR_ID
```

3. Start the DB Audit Management Console and connect to your database.

4. Click **Data Audit / Set Table and Column Aliases** menu.

5. Locate HR.EMPLOYEE table in the list of audited tables.

6. Enter V_HR_EMPLOYEE for the table alias and click the **Close** button to close the **Aliases** screen.

To test this custom-view based report:

1. Click **Reports > Data-change Audit Reports > Audit Trail by Table** menu.

2. When prompted for the table name, choose HR.EMPLOYEE.

3. If needed, specify optional reports parameters.

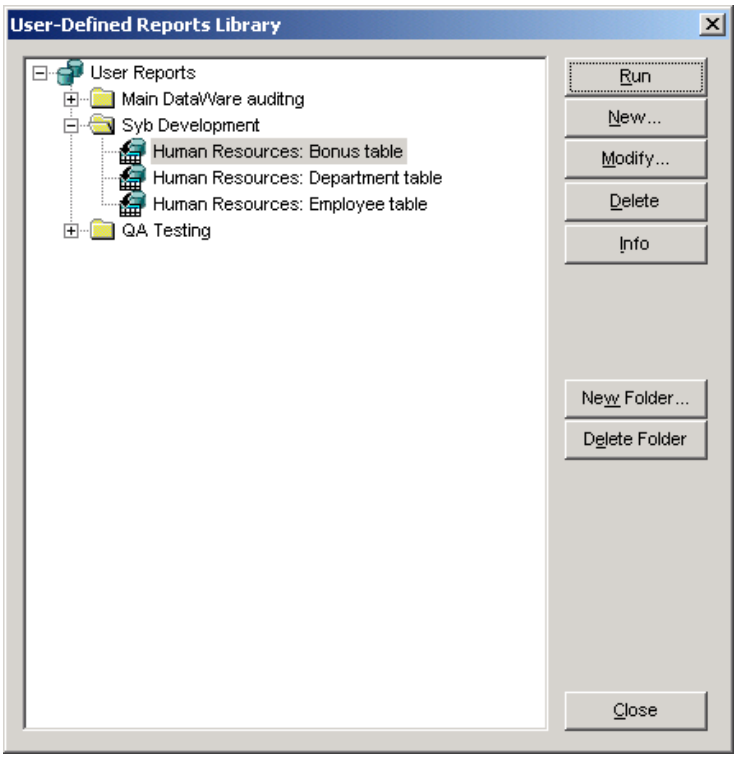
The report displayed in DB Audit Management Console is now based on the results returned from the view.

User-Defined Report Library

DB Audit allows you to quickly create your own custom reports for displaying results of data-change auditing using the built-in Custom Report Designer tool. New custom reports designed directly in the DB Audit Management Console are automatically added to the User-Defined Reports Library.

The User-Defined Reports Library is accessed from the Reports menu by selecting the **Reports > User-Defined Reports** menu item or by pressing **Ctrl+U** shortcut. The Report Library is stored, by default, in the **reports.dar** file. This file can be found in the DB Audit installation directory. You can freely copy this file between different computers or place it on a shared network drive where multiple DB Audit users can share it. You can use the DB Audit options (**File > Options** menu) to customize the location of this file.

Because the number of user-defined reports can grow quickly, DB Audit provides an option to organize these custom reports in logical groups that can be stored in folders created within the Report Library.



Working with the Report Library

Running a custom report

1. Open the User-Defined Reports Library.
2. Expand the folder containing the report you want to run
3. Double-click the report or highlight the report and click the **Run** button.

Renaming a report or folder

1. Open the User-Defined Reports Library.
2. Select the desired report or folder, then either press the **F2** key or click the highlighted label. Type the new label text and then press the **Enter** key.

   Note: Changing the report label does not change the report title displayed in the report header. To change both the title and the label, click the **Modify** button and then follow the instructions provided by the **Report Wizard**; refer to the Creating Custom Data-change Audit Reports topic for more information.

Moving a report to another folder

1. Open the User-Defined Reports Library.
2. Select the desired report and then drag the report to the destination folder in the Report Library.

Deleting a report

1. Open the User-Defined Reports Library.
2. Select the desired report and then click the **Delete** button.

Displaying report modification status and target database

1. Open the User-Defined Reports Library.
2. Select the desired report and click the **Info** button. DB Audit will display a message box with status information, including the name of the database in which the original report was created, the date and time when the report was created, and the time it was last modified, as well as the name of the user who created or last modified the report.

Deleting report folder

1. Open the User-Defined Reports Library.
2. Select the desired folder and then click the **Delete Folder** button.

 **Note:** Deleting a report folder will also delete all reports contained in that folder.

Creating new report folder

1. Open the User-Defined Reports Library.
2. Click the **New Folder** button. The **Enter Folder Name** dialog will appear. Type in the name of the new folder and then press the **OK** button.

Modifying an existing report

1. Open the User-Defined Reports Library.
2. Select the desired report.
3. Click the **Modify** button. Follow the instructions provided in the **Report Wizard. R**efer to the Creating Custom Data-change Audit Reports topic for more information.

Creating a new report

1. Open the User-Defined Reports Library.
2. Select the folder in which you want to create a new report.
3. Click the **New** button. Follow the instructions provided in the **Report Wizard**. Refer to the Creating Custom Data-change Audit Reports topic for more information.

Creating Custom Data-change Audit Reports

DB Audit provides a graphical Report Wizard utility that is used to create various user-defined reports in five easy steps or less. Some steps are optional. The Report Wizard is accessible through the Report Library either by clicking the **New** button on the Report Library dialog or by clicking the **Modify** button. The following paragraphs describe each step and all the available options in detail.

Step 1: Select audited table

This step is required. The Report Wizard displays the list of tables being audited. The list consists of two columns – Table Alias and Physical table name. For information on how to define table aliases, refer to the Setting Table and Column Aliases topic. You must select an audited table whose data-change audit trail will be analyzed by the new report. To select a particular table, click the checkbox in the left-most column in the table list.

Click the **Next** button to advance to the next step.

Step 2: Select columns, groups and lookup values

This step is required. The Report Wizard displays a list of columns available in the audit trail for the table selected in Step 1. Each column is displayed on a separate line. Each line contains two checkboxes, two buttons and two columns – **Column Alias** and **Physical** column name. The column aliases will be used later in headers of selected report columns. You must select at least one column for the report. To select or deselect a column, click the checkbox in the **Show** column. To quickly select or deselect all columns in the list, use the **Select All** or **Deselect All** buttons.

You can optionally group report data by values from one or more report columns. For example you may want to group data by user name. To define a new data group, place a checkmark for the desired group items in the **Group** column. For each group column, you can optionally specify an aggregate group expression which will be calculated and displayed in the *group totals* band at report runtime. Such group expressions are called group totals. To select a particular expression, click the right-most button  in the column list. The **Choose Expression** dialog will appear. Choose the desired expression type or choose *none* to clear the current expression.

 **Note:** Use the **sum** and **avg** expressions only with numeric columns. Using them with non-numeric columns will lead to run-time errors during report execution.

You can optionally specify lookup values to be used instead of data values recorded in the table audit

trail. For example, you may want to display descriptive department names instead of numerical department IDs. To define a lookup list for a particular column, click the second from right button  in the column list. The **Enter Lookup** values dialog will appear. You will be presented with options for manually entering lookup values, importing them from a tab-separated text file, or importing them from a database table.

Click the **Next** button to advance to the next step.

Note:
Although all audited columns are displayed in the column list and it is possible to select them for the report, the report cannot display columns with user-defined data types that do not map directly to the standard ANSI compliant data types at runtime. In addition to user-defined data types, the following data types native to the DBMS also cannot be displayed:

Oracle: BLOB, BFILE, CLOB, RAW, LONG RAW, object, VARRAY, nested table, and REF columns.

SQL Server, ASE, ASA: IMAGE and all data types that cannot be converted to VARCHAR using the built-in *convert* function.

Step 3: Specify report name

This step is required. The Report Wizard prompts you for the report name. Note that the report name is also used for the report title. Type the desired name. The report name can contain any characters including spaces.

Click the **Next** button to advance to the next step.

Step 4: Specify report filter and sort

This step is optional. The Report Wizard prompts you to specify a report filter and sort criteria that will be added to the report SQL query and applied at the back-end while executing the report. The SQL entered must be valid and must follow SQL syntax and rules understood by your database system. Failure to specify valid SQL will lead to report errors during runtime. If you don't want to specify a report SQL filter and/or sorting criteria, click the **Next** button to advance to the next step. The Report Wizard will create a sample report which you can then customize in step 5.

Step 5: Customize report design

This step is optional. You may customize report appearance by resizing and rearranging columns on the displayed report sample.

Click the **Finish** button to close the Report Wizard and return to the Report Library. The new report label will appear highlighted in the current Report Library folder.

Example: Creating a Data-change Audit Report in DB Audit

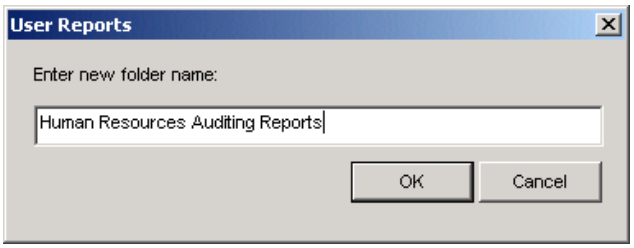
For this example, we will create a basic report to display the result of auditing of an Employee table. For this basic report, we will create a simple filter that displays only changes made in the Employee table by users whose primary department is not Human Resources.

1. **Open Report Library:**

   Click **Reports > User-Defined Reports** menu or press **Ctrl+U** shortcut. The Report Library dialog will appear.

2. **Create new folder:**

   Click the **New Folder** button. The User Reports dialog will appear. Type in *Human Resources Auditing Reports*. Press the **OK** button on the dialog.

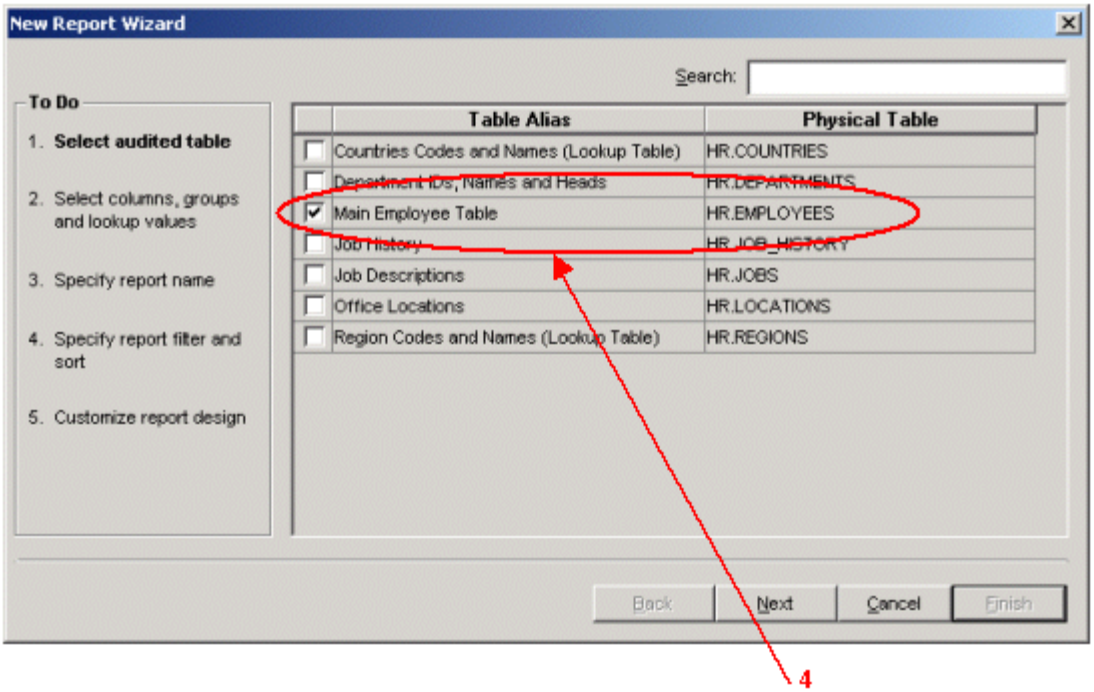


3. **Start the New Report Wizard:**

   Click the **New** button. The **New Report Wizard** will appear.

4. **Select the audited table:**

   Click the check box in the *Main Employee Table* row. Note that the name *Main Employee Table* is an alias for the HR.EMPLOYEES database table.



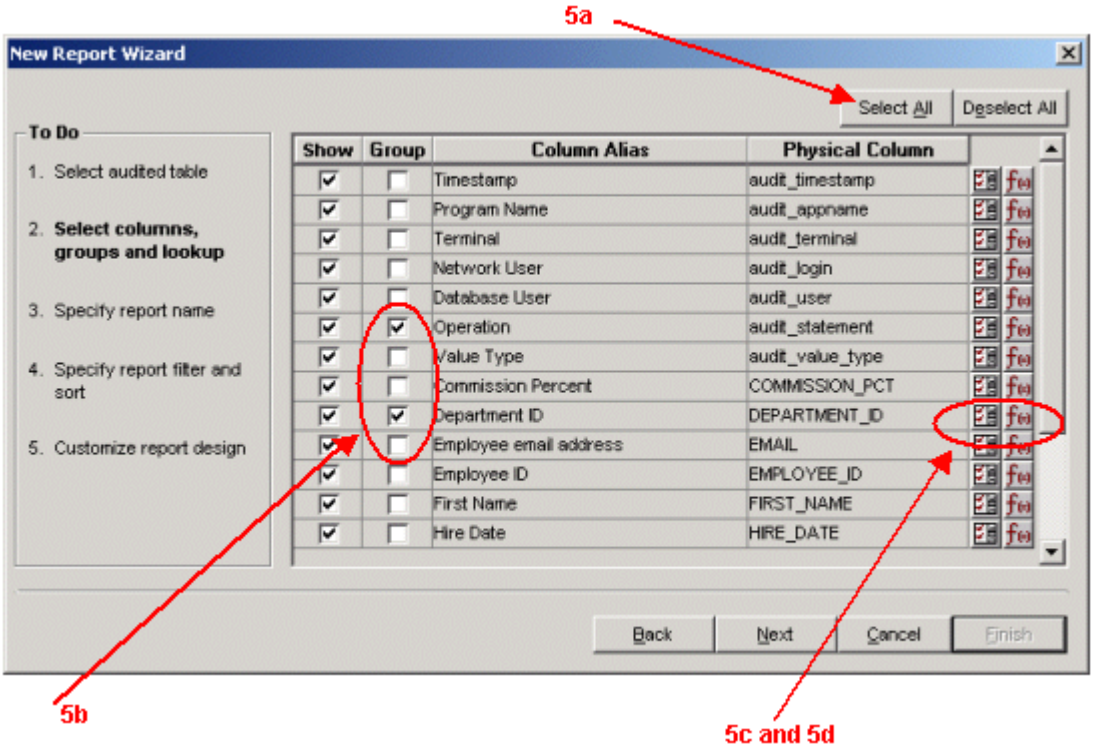
   Click the **Next** button.

5. **Specify columns, groups and totals:**

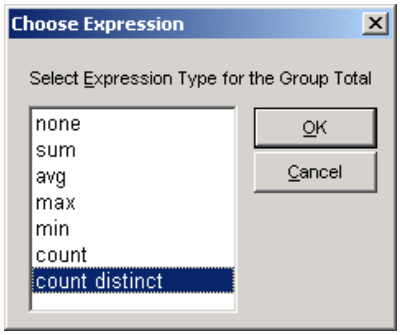
   For simplicity's sake, let's select all available columns for our sample report and group report data by type of change (INSERT, DELETE/UPDATE) and the affected department.

a. Click the **Select All** button to select all columns.

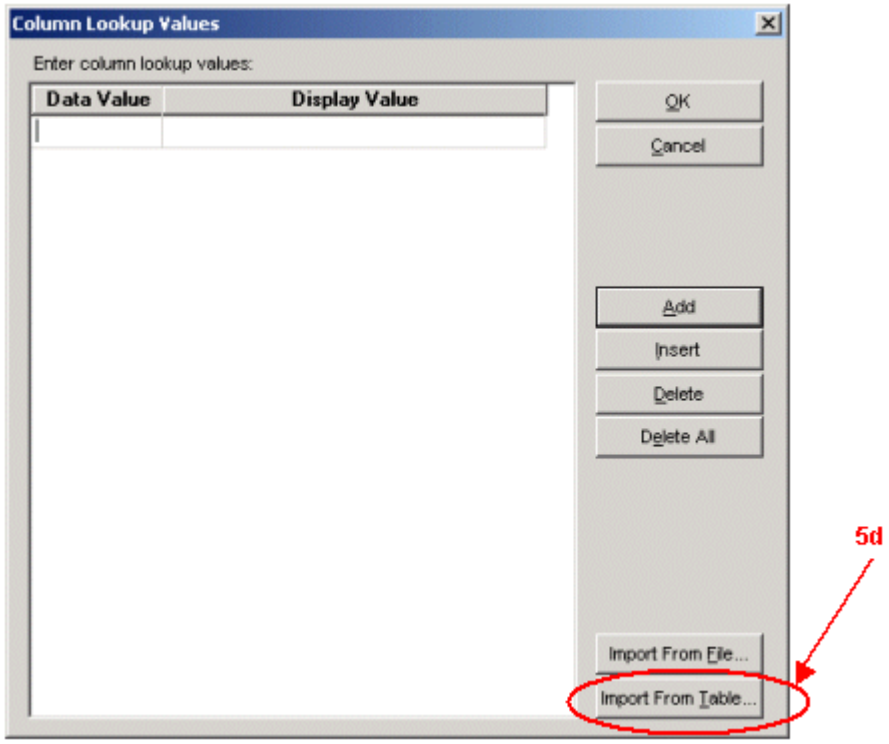
b. Click checkboxes in the **Group** list column for the *Operation* and *Department ID* table columns. This will allow report grouping by data from those two columns.



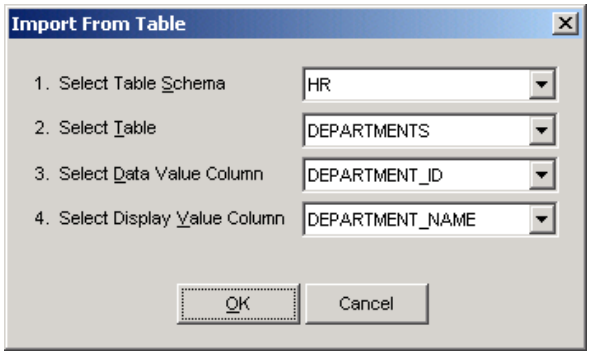
c. Click the Expression button displayed in the *Department ID* row. The **Choose Expression** dialog will appear. Double-click the *count distinct* expression type.



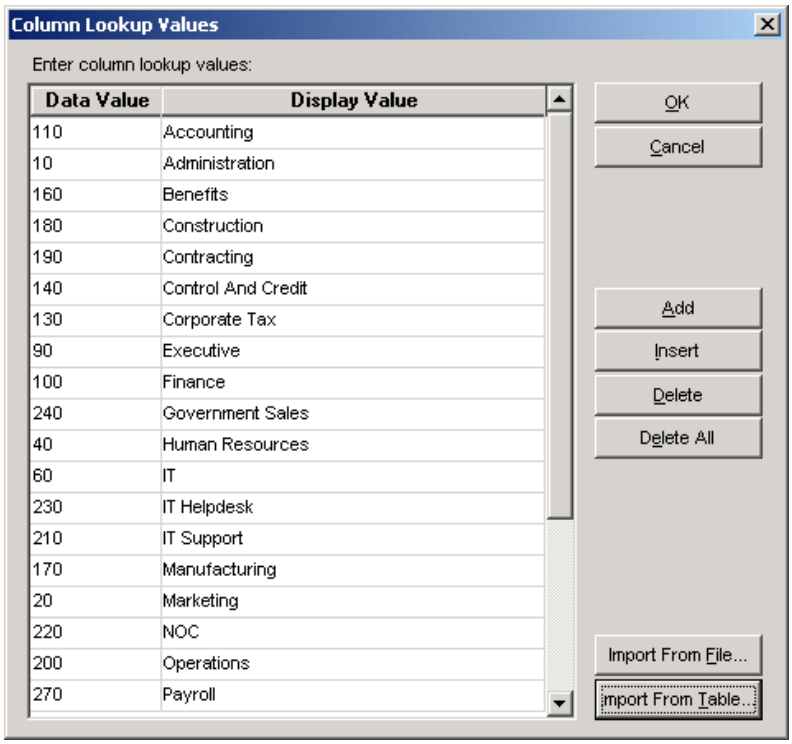
d. Click the Lookup Values button displayed in the *Department ID* row. The **Column Lookup Values** dialog will appear. Instead of displaying numerical department Ids on the example report, let's import department names from another table and show them on the report.



Click the **Import From Table** button. The **Table Import** dialog will appear.



Select *HR* for from the **Table Schema** drop-down.
Select *DEPARTMENTS* from the **Table Name** drop-down.
Select *DEPARTMENT_ID* from the **Data Value Column** drop-down.
Select *DEPARTMENT_NAME* from the **Display Value Column** drop-down.
Press the **OK** button. DB Audit will fetch data from the **HR.DEPARTMENT** table and populate the **Column Lookup Values** list as shown below.

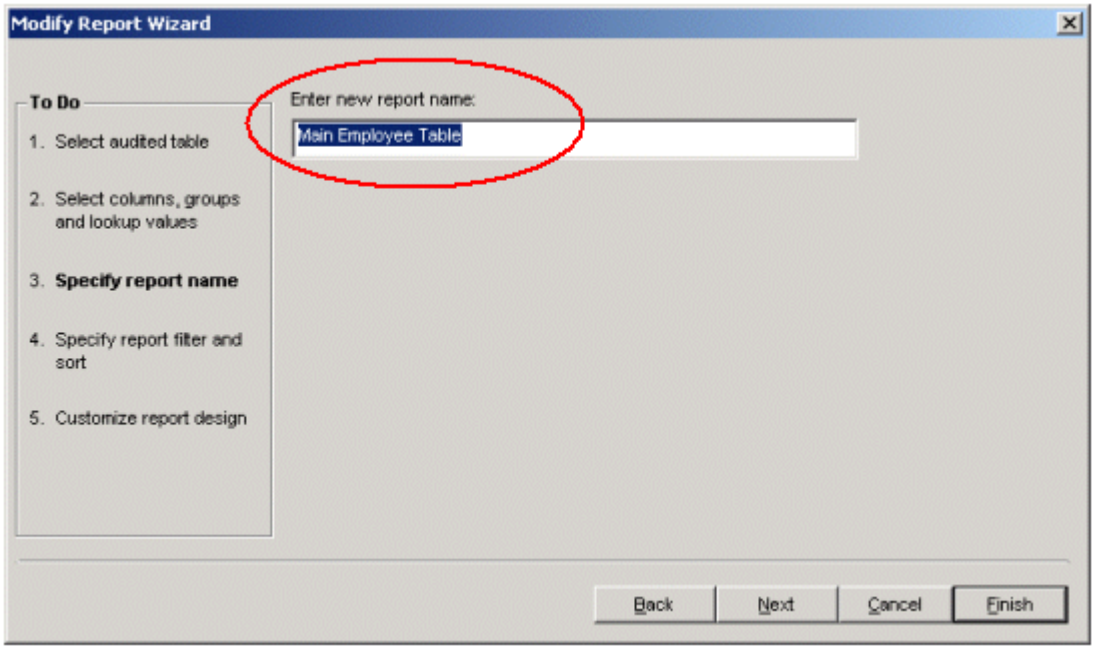


Click the **OK** button to close the dialog and return to the **New Report Wizard**.

Click the **Next** button on the **New Report Wizard** screen.

6. **Specify report name:**

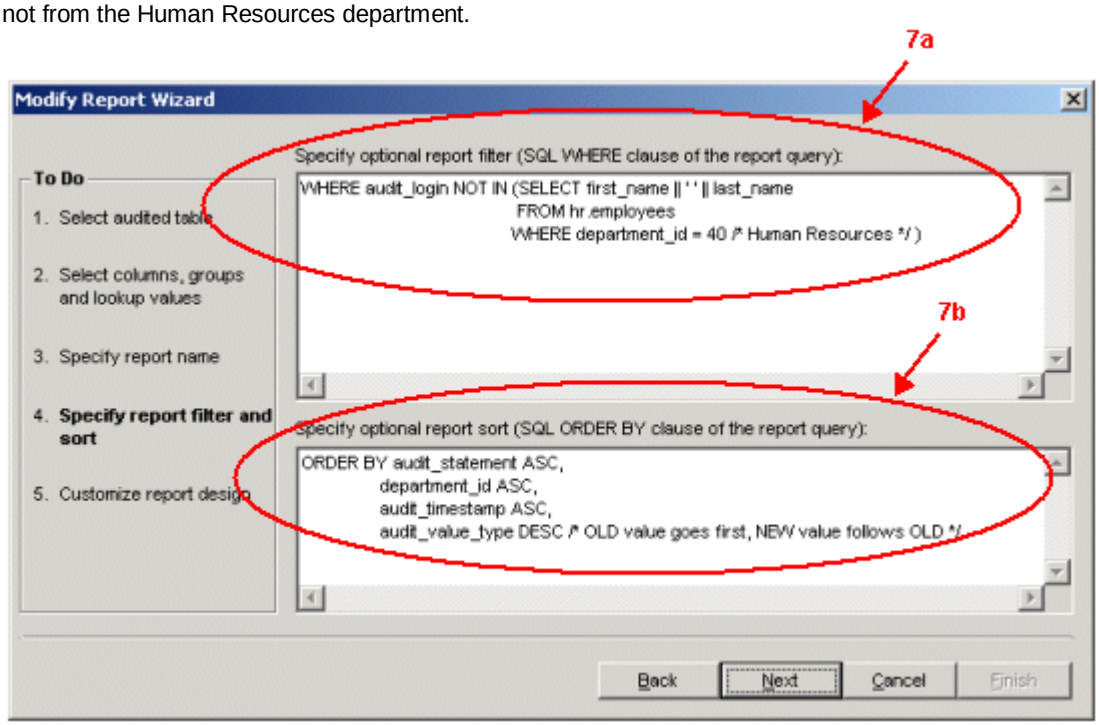
Note that the report name is also used for the report title. You can change the report name and title if required.



Click the **Next** button.

7. **Specify report filter and sort:**

   a. Now let's specify a report filter that will select only changes made by users who are not from the Human Resources department.



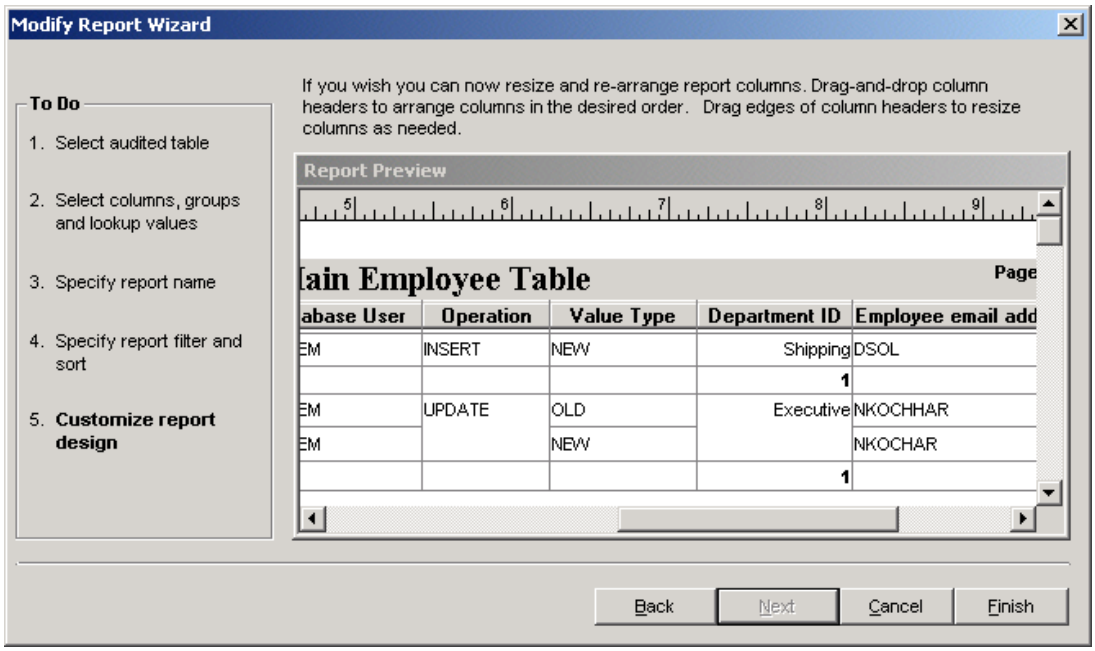
   In this example we assume that users log in to the corporate network using their full name as "*FirstName LastName"* so that their network login name can be compared against data in the employee table. The Filter definition specified in the SQL WHERE clause contains SQL that is used in the sub-query to return the names of people who are not working in the Human Resources department.

   b. Let's customize the default report sort order based on the selected group columns. Let's also sort the example report data by date and time (*audit_timestamp* column) and by audit value type (*audit_value_type* column). This will force all changes within report groups to appear in chronological order. The UPDATE type of changes forces the original values to always appear before the updated values.

   Click the **Next** button.

8. **Customize report appearance:**

   Resize report columns to optimally fit contents and allocate as little report space as possible. Because the example report contains many columns and spans multiple pages, let's reduce the number of pages in the report by squeezing less important columns and columns whose data normally contain just a few characters.



Click the **Finish** button to close the **New Report Wizard** and return to the Report Library dialog.

9. **Run sample report:**

   Click the **Run** button to try out the newly created example report.

Example: Creating a Data-change Audit Report in Microsoft Excel

To demonstrate this option, we will create a basic report to display the results of auditing of an Employee table in Microsoft Excel. For this basic report, we will create a simple filter that displays only changes made in the Employee table by user John.

Step 1: Find out the name of the audit-trail table for the Employee table

Use any appropriate database query tool to find out in which table the data-change audit trail is stored for the Employee table. For example, if you use Oracle SQL*Plus or IBM Command Center, connect to the database and then execute the following query

```
SELECT audit_table FROM db_audit.data_audit_trail WHERE source_table = 'EMPLOYEE'
```

If you use Microsoft SQL Query Analyzer or Sybase ISQL, connect to the database, set the current database to the audit repository database using available graphical options or using the USE [database] SQL command, and then execute the query above.

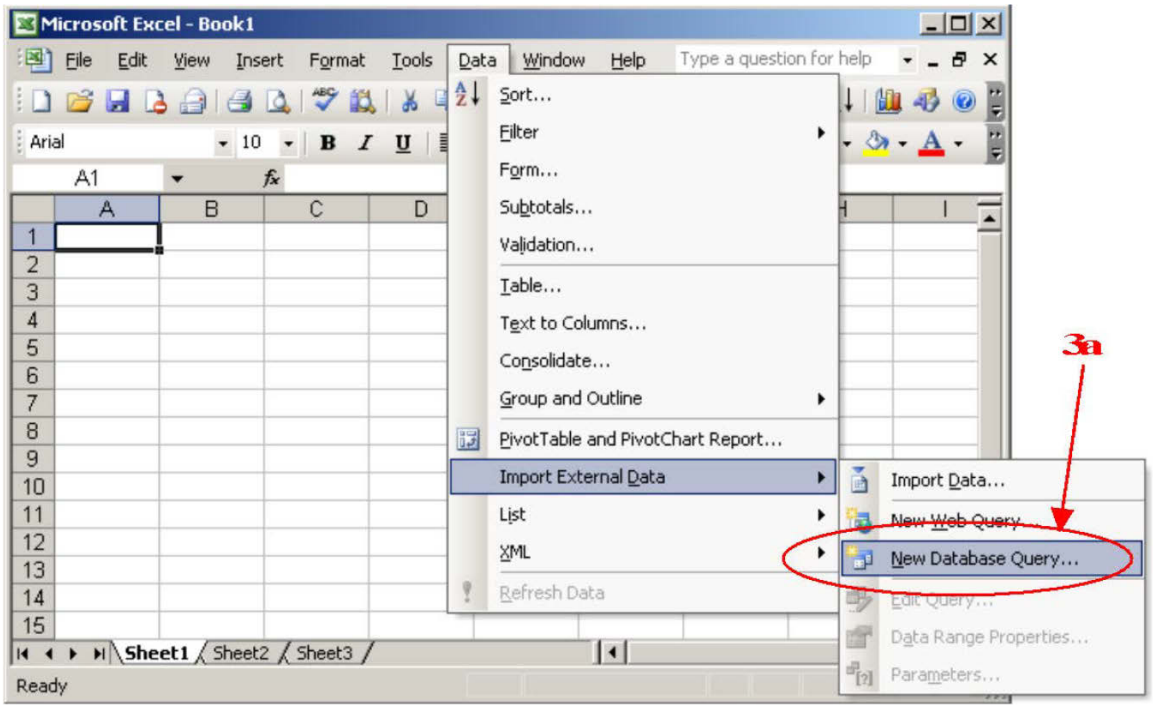
Step 2: Create a new Data Source

If you are going to create your first report and do not yet have an ODBC database connection configured for Excel, you must create a new ODBC Data Source for your database as instructed in CHAPTER 2, Defining the ODBC data source topic. For this example, let's call the new Data Source "SQLServer."

If you already have an ODBC connection configured, you can skip this step.

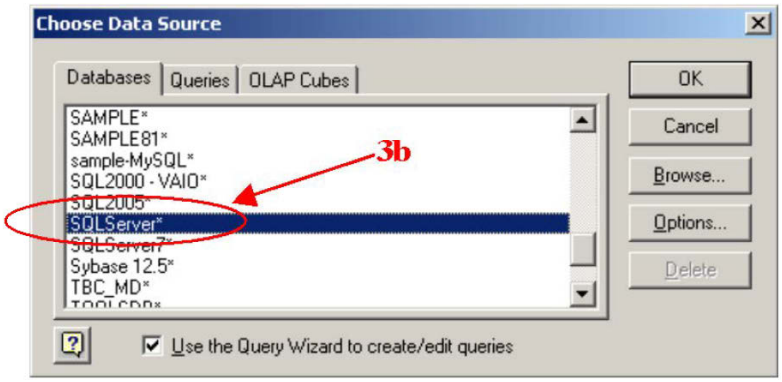
Step 3: Create new Data Query

a) Start Microsoft Excel and click the **Data > Get External Data > New Database Query** menu.

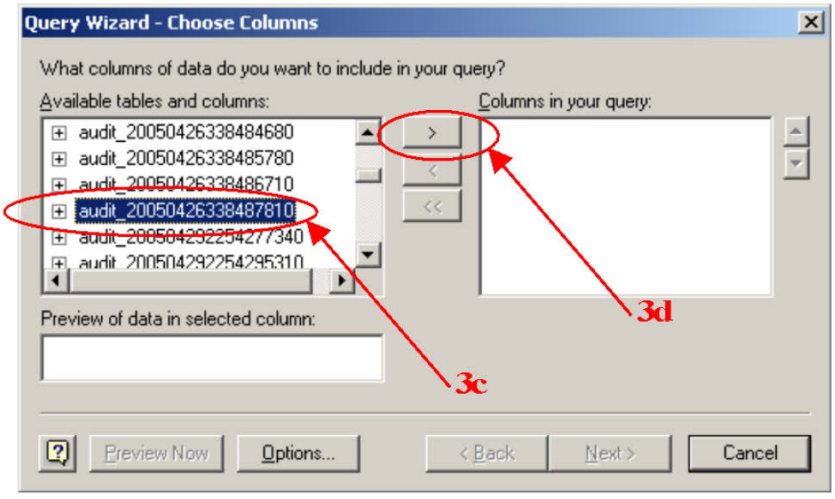


This will open **Choose Data Source** dialog.

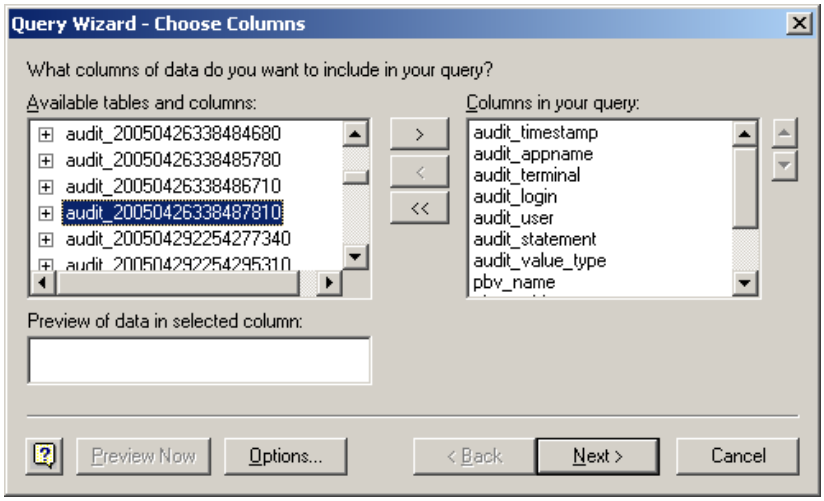
b) Select the ODBC Data Source created in step 2 and then click the **OK** button. This will start the **Query Wizard**.



c) In the **Available tables and columns** list, select the name of the appropriate audit trail table. This should be the name you obtained in step 1.

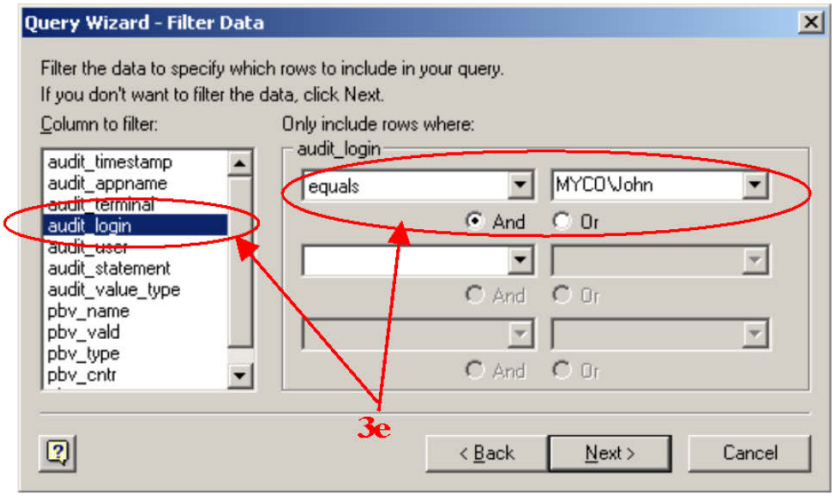


d) Click the ">" button to add all columns available in the selected table to the **Columns in your query** result set. This will add all columns to the Excel report. If you do not need all columns, click the "+" symbol next to the table name to expand the table listing and then double-click each column you want to add to your Excel report. The result should look like the following:



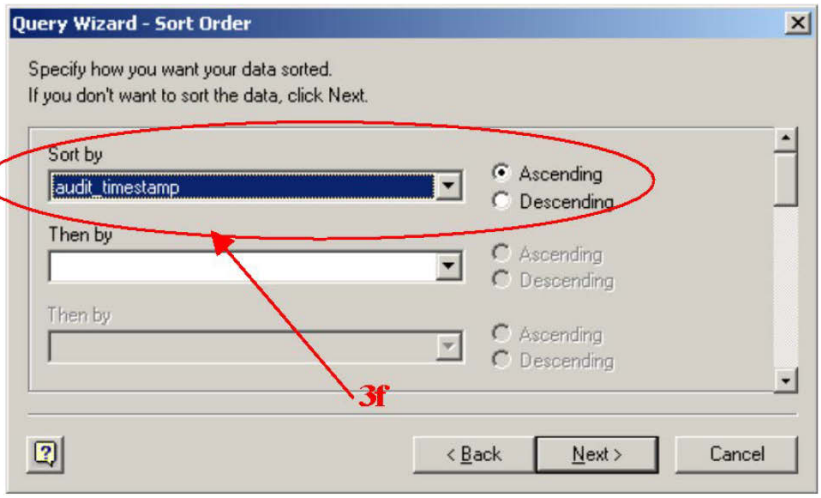
Click the **Next** button to continue.

e) Specify **Report Filter** on the query level. For example, click on the *audit_login* column name in the **Columns to filter** list and then in the **Filter** area, select "equals" from the drop-down list, then type *MYCO\John* for the login name. This specifies that only return audit records for activities performed by *John* should be returned.



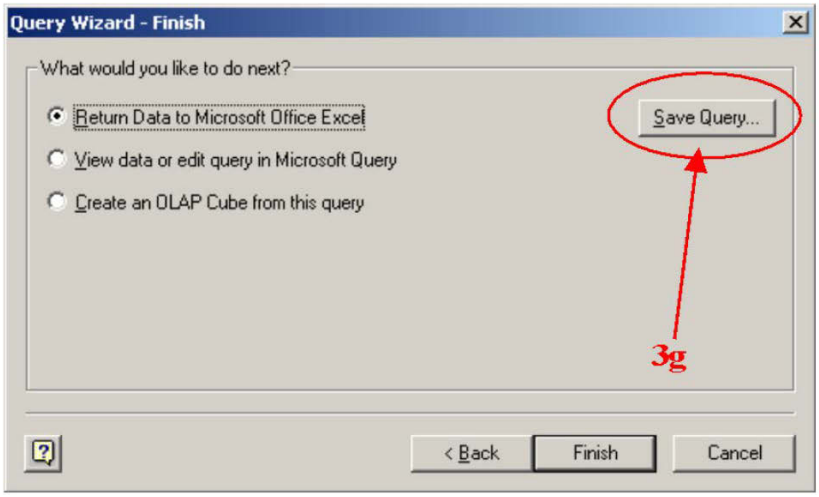
Click the **Next** button to continue.

f) Specify the sort order for report data at the query level. For example, choose *audit_timestamp* column name to sort the report chronologically by event time.



Click the **Next** button to continue.

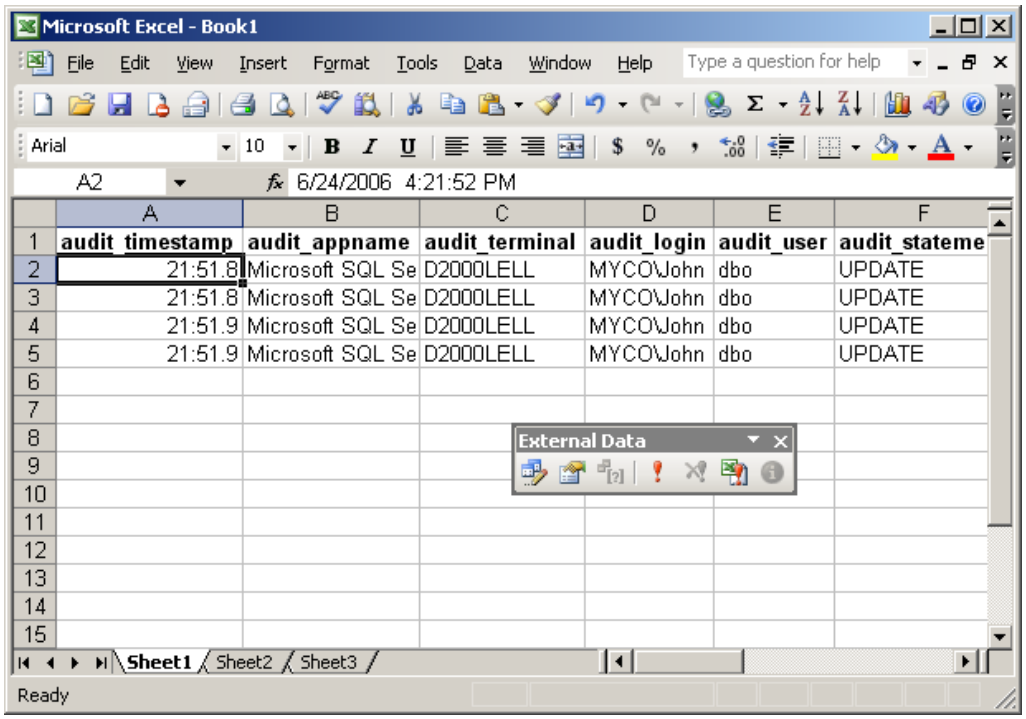
g) If you want to reuse the query for another report, click the **Save Query** button and, when prompted, enter a file name for the saved query.



Click the **Finish** button to finish the query design and return to Microsoft Excel.

Step 4: Select report placement and format, and customize audit data

After the Query Wizard closes, Excel will prompt you to specify where to insert the results of the Data Query. You can choose an existing worksheet or create a new one and you can also choose the starting cell. It is recommended that you use the default top left cell, which will place column headers in the first row. Then you can easily apply additional filtering and sorting rules. The resulting worksheet should like the following.



Use available Excel features to format and customize the report data as needed.

Example: Creating a System Audit Report in Microsoft Excel

To demonstrate this option, we will create a basic report to display the results of database user activity auditing in Microsoft Excel.

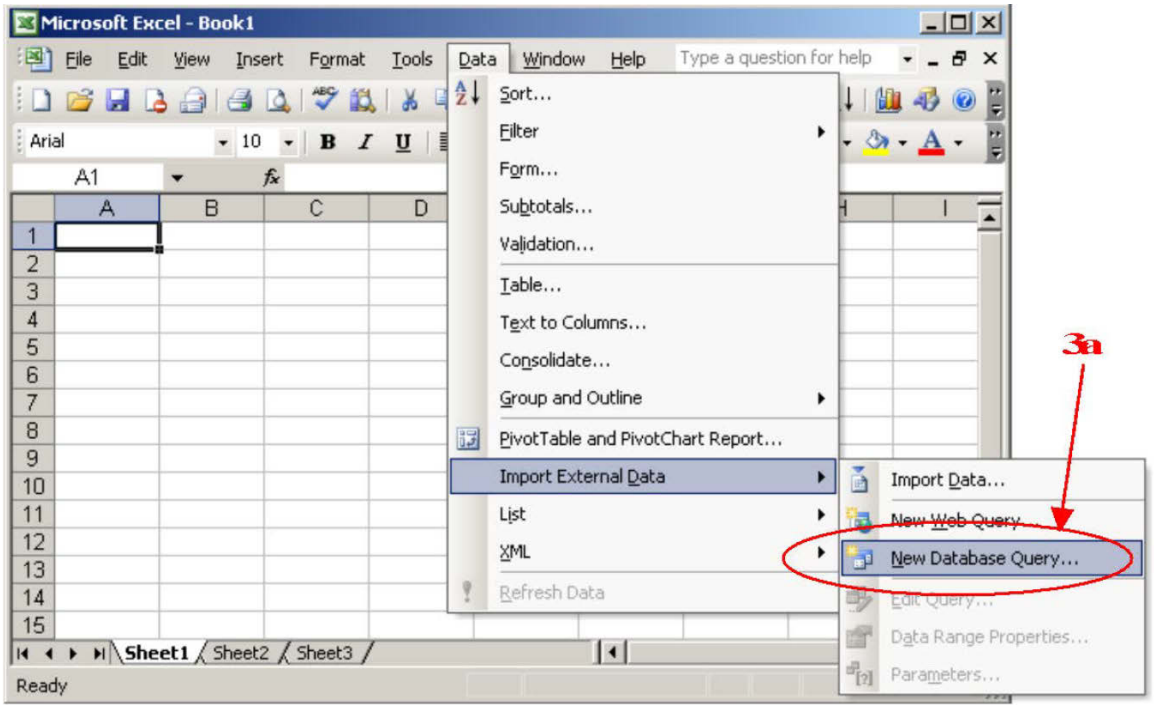
Step 1: Create new Data Source

source If you are going to create your first report and do not yet have an ODBC database connection configured for Excel, you must create a new ODBC Data Source for your database as instructed in CHAPTER 2, Defining the ODBC data source topic. For this example, let's call the new Data Source "SQLServer."

If you already have an ODBC connection configured, you can skip this step.

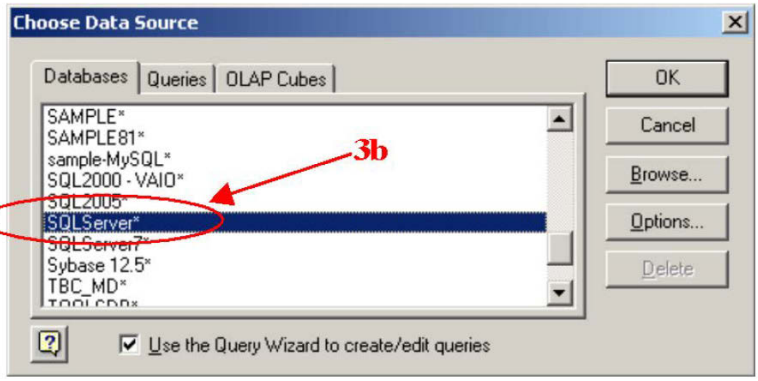
Step 2: Create new Data Query

a) Start Microsoft Excel and click **Data > Get External Data > New Database Query** menu.



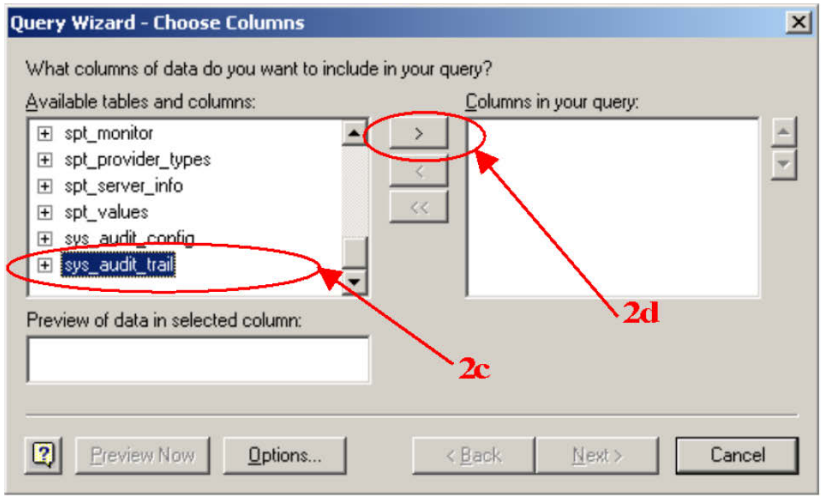
This will open **Choose Data Source** dialog.

b) Select the ODBC data source created in step 2, then click the **OK** button. This will start the **Query Wizard**.

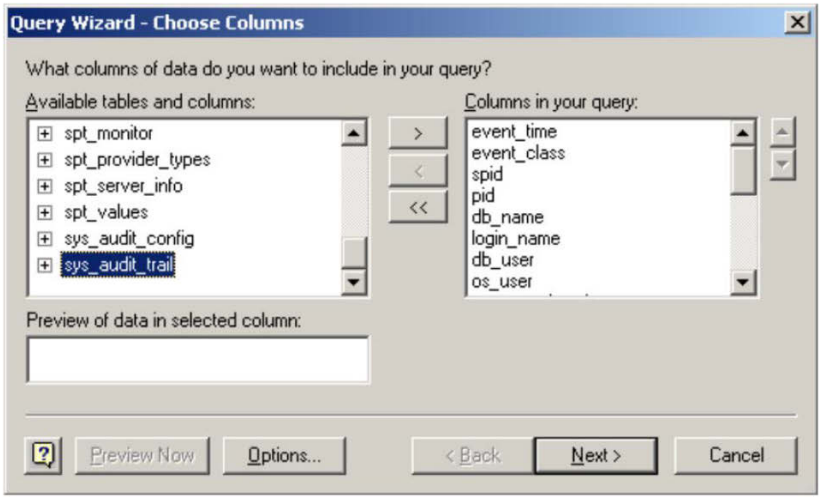


c) In the **Available tables and columns** list, select one of the following table names as appropriate for your DBMS:

- **SQL Server:** db_audit.sys_audit_trail.
- **Oracle (if alternate audit trail option is used):** db_audit.dba_audit_trail.
- **Oracle (if default audit trail table is used):** sys.dba_audit_trail.
- **DB2:** db_audit.sys_audit_trail.
- **ASE:** dbo.audit_XX – here XX must be replaced with the numeric suffix of the currently used audit table.

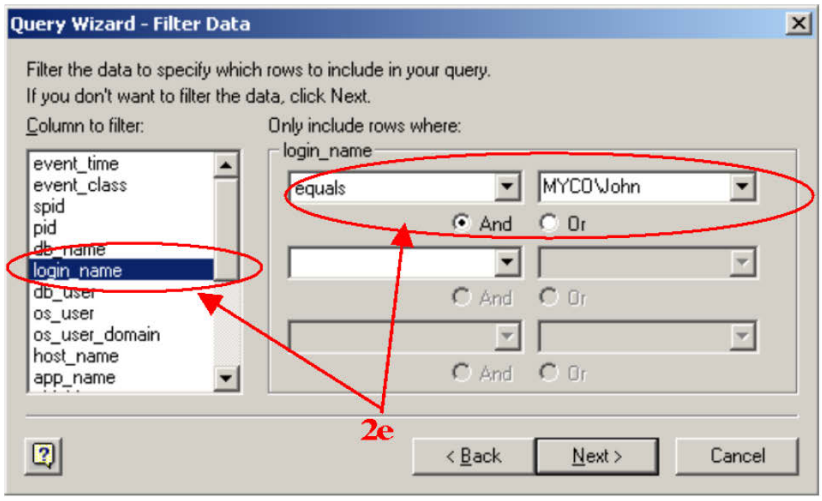


d) Click the ">" button to add all columns available in the selected table to the query result set and to include them in the Excel report. If you do not need all columns, click the "+" symbol in front of the table name to expand the list of columns in the table, then double-click each column you want to include in your Excel report. The result should look like the following image.



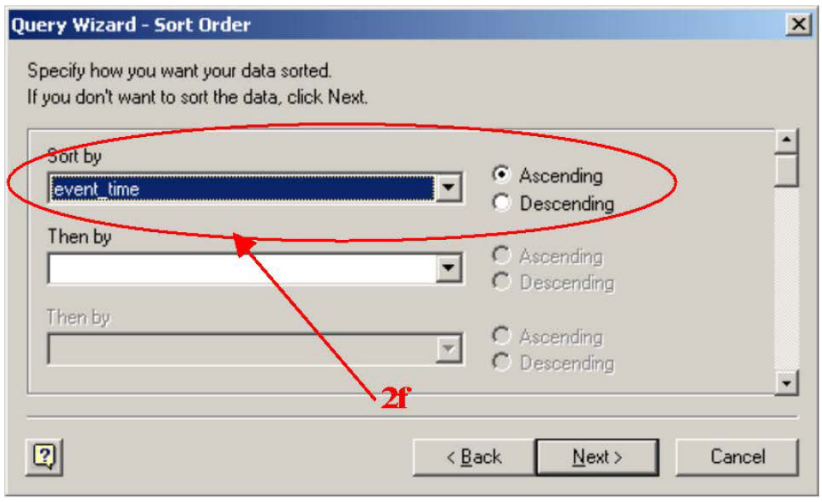
Click the **Next** button to continue.

e) Specify **Report Filter** on the query level. For example, click the *login_name* column name in the **Columns to filter** list, then in the **Filter** area, select "equals" from the drop-down list. Type *MYCO\John* for the login name as shown in the screenshot below. This will return only audit records for activities performed by *John.*



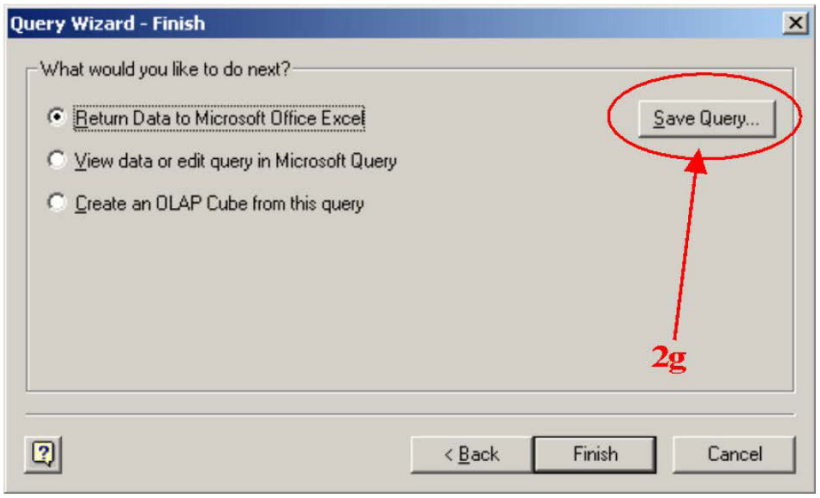
Click the **Next** button to continue.

f) Specify the sort order for report data at the query level. For example, click the *event_time* column name to sort the report chronologically by event time.



Click the **Next** button to continue.

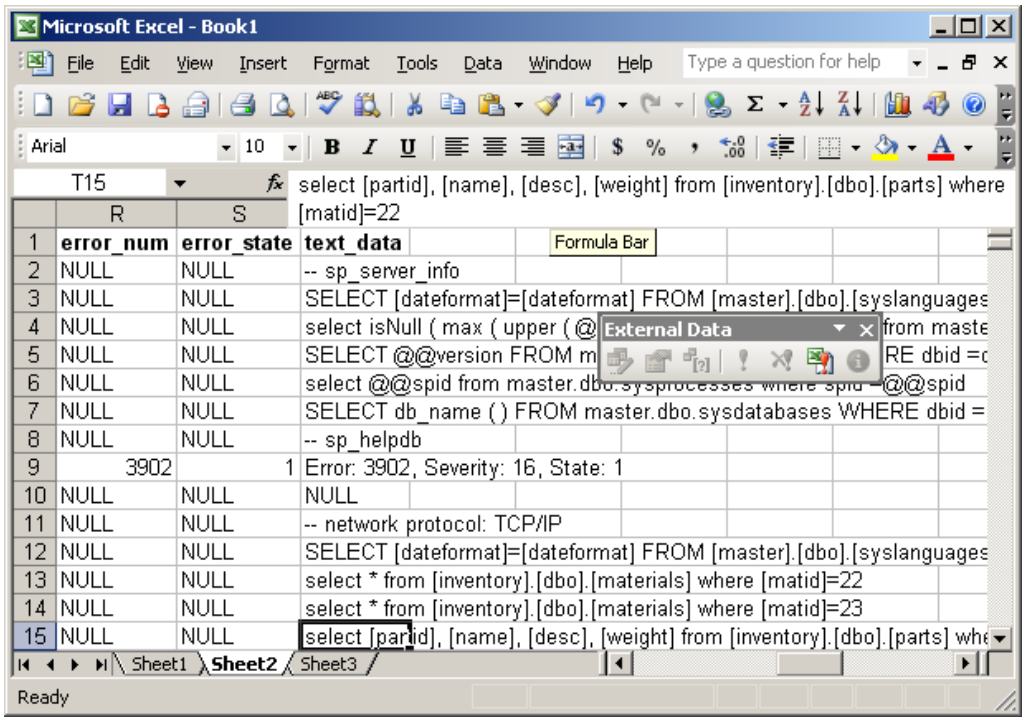
g) If you want to reuse the query for another report, click the **Save Query** button and, when prompted, enter a file name for the saved query.



Click the **Finish** button to finish query design and return to Microsoft Excel.

Step 3: Select report placement and format, and customize audit data

After the Query Wizard closes, Excel will prompt you to specify where to insert the results of the Data Query. You can choose an existing worksheet or create a new one. You can also choose the starting cell. It is recommended that you use the default top left cell, which will place column headers in the first row. Then you can easily apply additional filtering and sorting rules. The resulting worksheet should like the following.



	R	S	
1	**error_num**	**error_state**	**text_data**
2	NULL	NULL	-- sp_server_info
3	NULL	NULL	SELECT [dateformat]=[dateformat] FROM [master].[dbo].[syslanguages
4	NULL	NULL	select isNull (max (upper (@
5	NULL	NULL	SELECT @@version FROM m
6	NULL	NULL	select @@spid from master.dbo.sysprocesses where spid = @@spid
7	NULL	NULL	SELECT db_name ( ) FROM master.dbo.sysdatabases WHERE dbid =
8	NULL	NULL	-- sp_helpdb
9	3902	1	Error: 3902, Severity: 16, State: 1
10	NULL	NULL	NULL
11	NULL	NULL	-- network protocol: TCP/IP
12	NULL	NULL	SELECT [dateformat]=[dateformat] FROM [master].[dbo].[syslanguages
13	NULL	NULL	select * from [inventory].[dbo].[materials] where [matid]=22
14	NULL	NULL	select * from [inventory].[dbo].[materials] where [matid]=23
15	NULL	NULL	select [partid], [name], [desc], [weight] from [inventory].[dbo].[parts]

Use available Excel features to format and customize the report data as needed.

CHAPTER 8: Central Audit Repository

When to use a centralized repository

If you audit multiple databases, you should consider using a central repository for all your audit data.

Setting up a central repository has several advantages:

- A single repository can store and consolidate audit data from multiple database systems scattered across the enterprise. Audit data brought from different database systems is automatically converted to a common audit format convenient for analysis and reporting.
- The audit data is moved to a secure environment which is not accessible by the personnel working with the audited databases. This tamper proof setup eliminates the possibility of the audit data alterations.
- You can run alerts and reports analyzing all collected audit data from multiple servers in one place and get the big picture of your enterprise security.
- Running alerts and reports from the central repository doesn't require use of the processing resources of the audited database systems and thus lessens processing and performance requirements for these systems.
- Audit storage requirements in audited databases can be minimized because the collected audit data is moved to the central repository.
- Using a central repository lowers maintenance costs associated with managing and monitoring space usage in audited databases.

In addition to providing centralized location for audit trails and historical archiving of audit data, the central repository system can be used to store other multi-database security data including:

- Security Snapshots
- Historical results of PII, PCI, Banking Data Discovery
- Historical results of database and network vulnerabilities scanning
- Other security and audit data that must be stored in a place inaccessible to personnel whose activities are being audited.

Lastly, you can use DB Audit Alert Center server as a convenient platform for automating other data processing tasks as well as to automate loading of other security and application logs to the central repository server. From the central repository server, you can then generate aggregated reports connecting application logs, system logs, and database audit logs.

How it works

A central repository contains audit data from more than one database system. A central repository is not automatically created; to create a central repository, you must explicitly register your other database systems with the central repository. When you register a database system with a central repository, the central repository automatically replicates audit-trail data from the registered system. This means the following:

- Audit data from the registered database system is automatically added or moved to the

central repository.

- System audit data from different types of database systems is automatically transformed to the common format and is stored in a single audit trail.
- Audit data from data-change audit trails is stored in the central repository as a unique set of audit trail tables. This uniqueness is guaranteed. A unique suffix is added to each archived table. This allows audit data from two identical remote database systems having identically configured audit trail tables can be replicated to the central audit repository without conflict.

The Alert Center Scheduler periodically runs audit trail archiving jobs on remote databases. Once the archived data is archived to the central repository, it is purged completely or partially from the remote system.

By default, audit data archiving jobs are scheduled to run every 15 minutes, although you can change the scheduling frequency using the scheduler's graphical interface. The scheduler maintains a separate job queue for every remote database. This allows it to archive audit data from different systems concurrently while at the same time maintaining archiving order for each system.

Supported database systems and audit-trail archiving methods

DB Audit supports central repositories managed by the following database systems:

- Microsoft SQL Server 2000 with SP4 and later
- Oracle 9i and later
- IBM DB2 7.2 and later
- MySQL 5.0.27 and later

Audit data can be archived from any database system supported by DB Audit's system or data-change auditing. Audit data archiving is **not limited** to the systems listed above.

The repository can be installed on any Unix, Linux or Windows system supported by the database systems listed above.

Tips for using central repository in heterogeneous environments:

- Microsoft SQL Server provides more choices for the audit trail archiving methods then other database systems and thus provides greater flexibility for how the central repository can be setup and used. See the following paragraphs for more information.
- Object naming in Microsoft SQL Server is more flexible then in Oracle and in DB2. SQL Server-based repositories allow you to preserve names of data-change audit trails, while current name length limitations in Oracle (30-characters) and DB2 (8-characters) may cause name truncation which could potentially lead to naming conflicts.
- If you are not auditing heterogeneous database systems, it is always preferred to use the same type of the database for the central repository as you use for other database systems. This will ensure data type compatibility across all systems and will simplify system maintenance.

DB Audit currently supports the following audit trail archiving methods:

- **Data Pump** – Audit data is retrieved record by record from the remote database and inserted into the central repository database. This process is incremental; it only affects audit records generated in the remote database since last archiving.

  This method is not the fastest archiving method, but it is compatible with all database systems.

- **Bulk Load** – Audit data is retrieved in batches and saved to intermediate files which are then loaded into the repository using native database bulk load utilities or commands. The BCP utility is used with SQL Server repositories, the SQL*Loader utility is used with Oracle, and the LOAD command is used with DB2. This process is incremental; it only affects audit records generated in the remote database since last archiving.

  This method is fast but is not guaranteed to work with all database systems.

- **Linked Servers -** Audit data is retrieved from the remote database and inserted into the central repository database in a single pass using linked servers and the INSERT…SELECT command. This process is incremental; it only affects audit records generated in the remote database since last archiving.

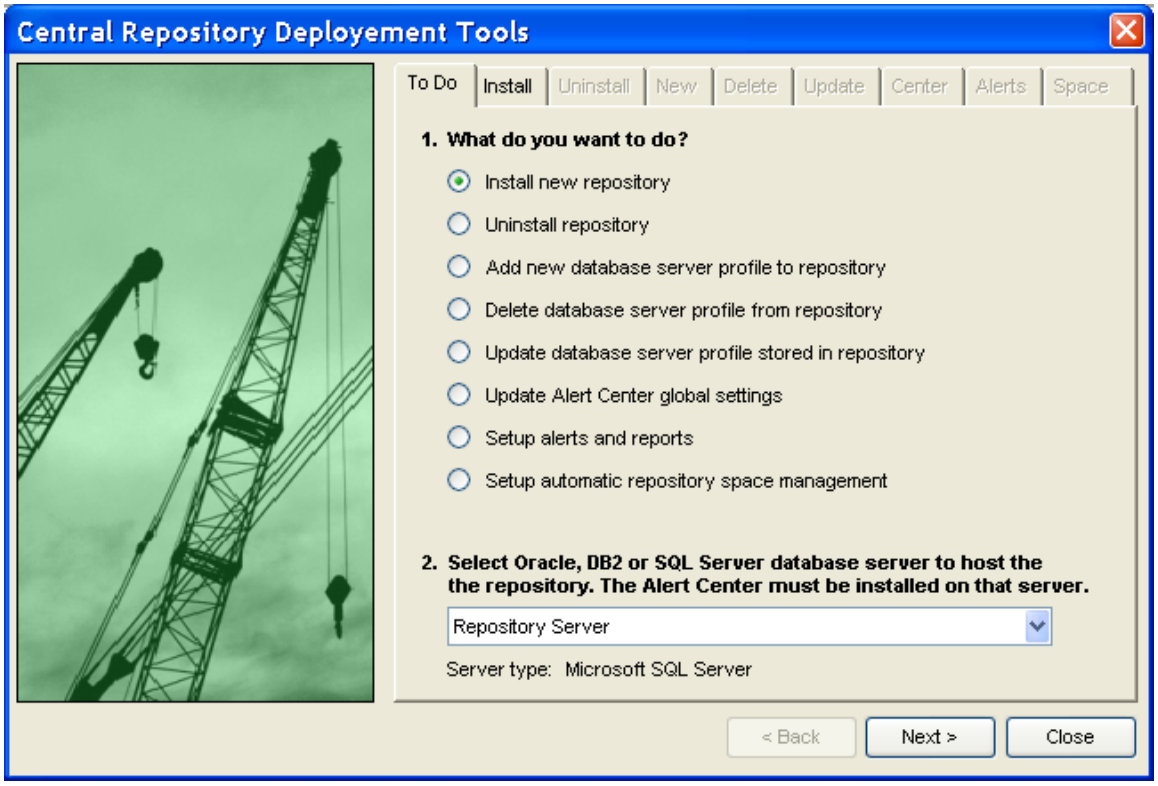
  This method relies on Microsoft SQL Server's Linked Server feature and can only be used with repositories hosted In Microsoft SQL Server.

All supported archiving methods pull audit trail data from remote databases into the central repository database. If an archiving operation is successful, the audit data can be then optionally erased in remote databases either partially or completely. When configuring the process, you can specify whether to partially or completely purge the remote audit trail and you can specify the number of days worth of data to leave on the remote database for compliance reporting.

Tip: Some Compliance Reports cannot be run off the central repository database because they require access to system dictionary tables in remote databases when analyzing the audit trail data. If you run these reports, you should not set up the audit data archiving jobs to completely erase remote audit trails after archiving. It is recommended that these jobs be set up to leave two or three months of the most recent audit data on remote systems for reporting purposes.

Installing, configuring and uninstalling a central audit repository

The central repository can be installed, configured and uninstalled using the Central Repository Deployment Tools which can be started from the DB Audit Management Console. In the DB Audit Management Console, click **Tools** menu then click **Central Repository Deployment Tools** menu item to launch the tools. The Central Repository Deployment Tools screen will appear.



The following prerequisites must be met to use the Central Repository Deployment Tools:

- The DB Audit Management Console must be installed on your workstation.
- Java Run-time Environment (JRE) or Java Development Kit (JDK) version 1.4 or better must be installed on your workstation. If you don't have the proper version of JRE or JDK installed on the system, visit Sun Microsystems web site http://java.sun.com where you can you can freely download JRE and JDK software.
- The JAVA_HOME system environment variable must be defined on your computer. This variable must point to the directory where your JRE or JDK is installed; for example, *C:\Program Files\Java\j2re1.4.2_05*.
- The Alert Center server must be installed and running on a computer in your network. See Alert Center Server Installation topic in CHAPTER 15 for more information on how to install and configure the Alert Center server.
- A network connection must be also available between the Alert Center computer and the computer running DB Audit Management Console.

Installing a new central repository

1. In the Central Repository Deployment Tools screen, select the **Install new repository** option.

2. In the **Repository host** drop-down box, select the database profile of the database that you want to use for the repository. The database profile connection for the central repository must be

already configured. See CHAPTER 2: Connecting to Your Database for more information on how to configure database connections.

Click the **Next** button to continue or click the **Install** tab page.

3. Skip this step if the database system selected in the previous step is not Microsoft SQL Server

   If the database system selected in the previous step is Microsoft SQL Server, you must choose the database in which you want to create the new repository.

   Important Note: If you are already auditing the repository server or plan to audit it, DO NOT select the same database you are using for the local audit trail to also host the central repository database.

4. Choose the **archiving scope**; that is, what you want to store in the central repository. You can choose one of the following:

   - **System audit trail** – use this option to archive the system audit trail only.
   - **Data-change audit trails** – use this option to archive data-change audit trail only.
   - **System and data-change audit trails** – use this option to archive both system audit trail and data-change audit trail.
   - **Other** – use this option to register servers with the repository system without actually archiving any audit data. Use this option only if you need to register servers for PII, PCI, Banking Data Discovery or for Security Snapshots but do not run any audit data archival processes.

5. In the **Servers** box, click on the left most check box to select servers that you want to register with the central repository system.

   Tip: If you want to use a different archiving scope for different servers, choose only the servers that match the selected scope (defined in step 4). You can register other servers later using the **Add new database server** option and select a different scope option at that time. Also, you can also always modify the scope later using the **Update database server** option. Read Adding, removing, and updating server registration in the central repository topic for more information.

   Use the drop-down list in the **Method** column to select desired archiving method for each server. Descriptions of supported archiving methods are available in this chapter in Supported database systems and audit-trail archiving methods topic. If you want, you can choose different methods for different systems.

   Use the drop-down list in the **Keep Days** column to select how many days of data to leave in local repositories of audited database systems after each archiving. If you select a positive number, the archiving job will automatically purge all audit records older than the specified number of days. If you select the **All** option, everything is purged from the trail except new audit records recorded during the archiving job run. If you select the **None** option, the purge is not performed and all audit records are left in local repositories of audited database systems. If you want, you can choose different purge values for different systems.

   Tip: You can choose different purge rules for the central repository than those you apply to audited database systems. It is recommended that you choose relatively short audit history for local audit trails (from 30 to 100 days) and a long history for the central repository (one year or longer). This way you can still run compliance reports on audited database systems without having to worry about audit data occupying too much space.

   Please note that some Compliance Reports cannot be run off the central repository database because they require access to system dictionary tables in remote databases when analyzing

the audit trail data. If you run these reports, you should not setup the audit data archiving jobs to completely erase remote audit trails after the archiving. It is recommended that you setup these jobs to leave two or three months of the most recent audit data on remote systems for reporting purposes.

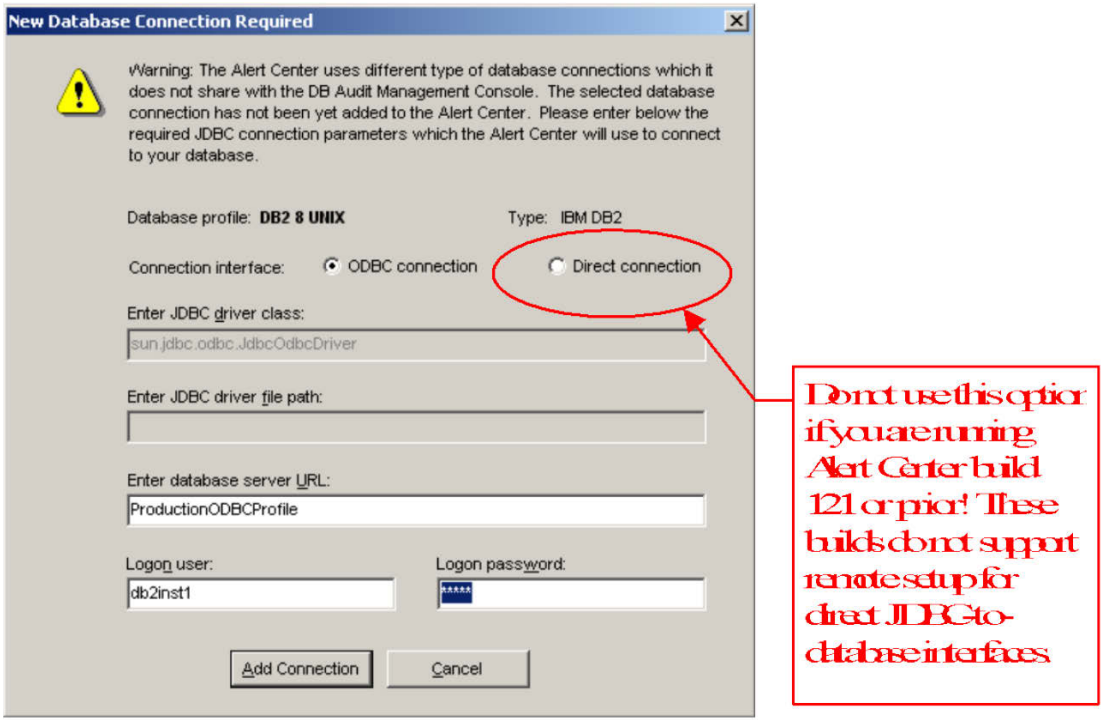
Click the **Next** button to continue or click the **Center** tab page.

6. Configure Alert Center connection to the central repository.

   Click the **Modify** button under "Connection profile name" to bring up the Alert Center database connection profile dialog **New Database Connection Required**.

   Notes: Do not confuse database profiles used by the DB Audit Management Console with database profiles used by the Alert Center server. These are different profiles used by different programs that could be running on different computers and using different connection methods.

   The **New Database Connection Required** dialog allows you to specify connection interface and connection properties for the Alert Center server only!



   On the **New Database Connection Required** dialog

   7. Select the **Direct Connection** or **ODBC connection** option as the Connection interface.

   8. If you selected the **Direct connection** type, enter placeholders for the database server name and port created for you in the "**Database Server URL**" input box.

      Note: Please make sure not to leave brackets indicating positions of placeholder; for example, if the placeholder appears as [SERVER NAME], make

sure you replace the entire placeholder, including [] brackets with the actual server name.

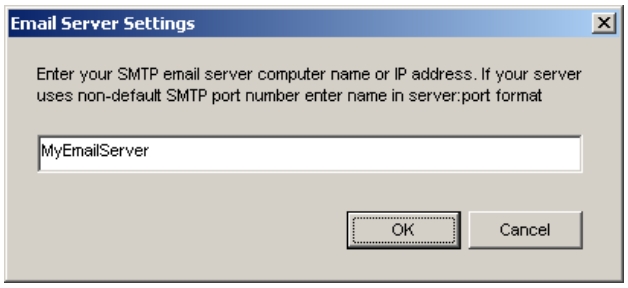
If you selected the **ODBC connection** type, enter name of an existing ODBC profile into the "**Database Server URL**" input box.

Note: Please make sure you enter ODBC profile name available on the computer running the Alert Center, not the one on the local computer from which you are remotely configuring the new alert and new database connection.

9. If you are using a non-trusted connection, enter database server logon and password. The specified logon must have permissions to execute SELECT type SQL queries on the system audit tables. For information on the system audit trail location and table names see Configuring System Audit Options topics in CHAPTER 3.

10. Click the **Add Connection** button to add the specified connection.

Click the **Modify** button under "Email server name" to configure the email server name. This will display **Email Server Settings** dialog as shown below.



This dialog allows you to enter name or TCP/IP address of your email server. If you don't want to change the current server name, click the **Cancel** button to close the dialog; otherwise click the **OK** button to save new settings and close the dialog.

Tip: By default the Alert Center uses the default SMTP port number 25 assigned to SMTP protocol. If your email server is set to use a non-default SMTP port number, specify the email server name in **server:port** format. For example, *myserver:125* or *192.168.0.50:125*.

You have now completed all required settings. Click the **Next** button to begin the Central Repository installation process.

Updating central repository settings

After the central repository is installed, you can change the following settings:

1. Modify registrations of audited database systems, their archiving methods and parameters. You can add new registrations or modify and delete existing registrations as needed. See Adding, removing, and updating server registration in the central repository topic later in this chapter for specific instructions on how to perform these tasks.

2. Modify email server settings. You can use any of the following methods to change these settings:
    1. Using the Alert Center Remote Console. See Configuring Alert Server Email Settings topic in CHAPTER 6 for detailed instructions.
    2. Using the Central Repository Deployment Tools. Start the Central Repository Deployment Tools; chose the Update Alert Center Global Settings option and then click the **Modify** button under "Email server name" to change the email server name. See previous topic for detailed instructions.
    3. Using the graphical interface of the Alert Center Scheduler. This method can be used if the scheduler is run in the graphical mode. While in the scheduler click **Tools > Options** menu to open the **Options** screen and then modify the email settings as required.
3. Modify report generation options and settings. You can use any of the following methods to change these settings:
    1. Using the Alert Center Remote Console. See Configuring Report Generation Options topic in CHAPTER 6 for detailed instructions.
    2. Modifying the AC.PROPS file directly on the server as described in Run-time Report Controls topic in CHAPTER 7.
4. Using the Alert Center Remote Console. You can add, modify and enable/disable central repository based alerts and reports. See Alert Center Remote Console topic in CHAPTER 6 for detailed instructions.
5. Fine-tune audit trail archiving and purging jobs. This can be performed using the graphical interface of the Alert Center Scheduler directly on the Alert Center server. Normally direct access to the Alert Center Scheduler is not required. Do not use this method unless you are instructed to do so by technical support.

Uninstalling a central repository

1. Launch the Central Repository Deployment Tools. Select **Uninstall repository** option.
2. In the **Repository host** drop-down box, select database profile of the database that you are using as the central repository and then click the **Next** button to continue or click the **Uninstall** tab page.
3. Choose **Yes** when prompted to confirm the uninstallation.
4. Click the **Next** button to begin the Central Repository uninstallation process.

    Note: The uninstallation process will drop previously installed repository objects and data, and will also remove DB_AUDIT user and login information from the repository database. Any associated alerts, reports and audit trail archiving jobs will be also removed.

Adding, removing, and updating server registration in the central repository

Important Notes:

- You should run the "Add Server" and "Update Server" functions on the computer from which the central repository is installed. During repository installation and updates, DB

Audit saves certain configuration information in the local system registry, and it needs this information later when updating the repository settings. Updating the central repository using multiple computers and non-unique database profile names may lead to invalid repository configurations containing duplicate references for same physical database servers.

- You should run the "Update Server" function after each major database server upgrade. Different major versions have different audit implementations, and the central repository needs to know which implementation is currently being used. Failure to update server registration in the central repository may lead to failed audit data archival and therefore to lost audit trail data.

Registering a new server

1. Launch the Central Repository Deployment Tools. Select **Add new database server profile to repository** option.
2. In the **Repository host** drop-down box, select the database profile of the database that you are using as the central repository and then click the **Next** button to continue or click the **Add** tab page.
3. Select the database server profiles that you want to register with the central repository. Select the archiving scope, methods and parameters for selected servers. For description of supported methods and parameters, see steps 4 and 5 in "Installing a new central repository" topic in this chapter. Click the **Next** button to begin the Central Repository update process.

   Tip: You can select multiple servers to be registered at once. If a server is already registered in the repository, its existing archiving settings will be automatically updated.

Updating registration of an existing server

1. Launch the Central Repository Deployment Tools. Select **Update database server profile stored in repository** option.
2. In the **Repository host** drop-down box, select he database profile of the database that you are using as the central repository and then click the **Next** button to continue or click the **Update** tab page.
3. Select the database server profiles whose registrations and archiving settings you want to update. Select archiving scope, methods and parameters for selected servers. For description of supported methods and parameters, see steps 4 and 5 in "Installing a new central repository" topic in this chapter. Click the **Next** button to begin the Central Repository update process.

   Tip: You can select multiple server registrations to be updated at once.

Removing an existing registration

1. Launch the Central Repository Deployment Tools. Select **Delete database server profile from repository** option.
2. In the **Repository host** drop-down box, select the profile of the database you are using as the central repository and then click the **Next** button to continue or click the **Delete** tab page.
3. Select database server profiles whose registrations you want to remove. Click the **Next** button to begin the Central Repository update process.

   Tip: You can select multiple server registrations to be removed at once.

Central repository audit trail space management

DB Audit Expert provides a simple and efficient method for controlling the size of audit trail data in a central repository system. Using the Alert Center you can schedule a job to run daily and purge old audit records from all audit trail tables. Use the following steps to schedule a new purge job or to update settings of a scheduled job.

1. Launch the Central Repository Deployment Tools. Select **Setup automatic repository space management** option.
2. In the **Repository host** drop-down box, select database profile of the database that you are using as the central repository and then click the **Next** button to continue or click the **Space** tab page.
3. In the **Days** field, enter the number of days of recent audit trail data you want to store in the audit trail tables.
4. In the **Time** field, enter time when you want the purge job to run.
5. If for whatever reason you do not want the job to run automatically, uncheck the **Enable purge job** checkbox.
6. Click the **Next** button to begin the Central Repository update process.

Configuring central repository based alerts and reports

DB Audit Expert supports two methods for adding central repository based alerts and reports. You can use the Central Repository Deployment Tools to quickly add or delete preconfigured alerts and reports to or from the central repository. You can use the Alert Center Remote Console to add, update and delete both preconfigured and custom alerts and reports, as well as to review alert and report processing logs and history.

Read the Alert Center Remote Console topic for detailed instructions on how to perform these tasks using the Alert Center Remote Console.

To use the Central Repository Deployment Tools to quickly add or delete pre-configured alerts and reports:

1. Launch the Central Repository Deployment Tools. Select **Setup alerts and reports** option.
2. In the **Repository host** drop-down box, select database profile of the database that you are using as the central repository and then click the **Next** button to continue or click the **Alerts** tab page.
3. In the Alerts and Reports list, check the items you want the Alert Center to run and uncheck items that you do not want to run.
4. In the **Recipient's Email** input field, enter the email address of the person or distribution group you want to alert when problems occur and to whom you want selected reports to be sent.

   If your email server requires user authentication, fill in **Sender's Email/Name** and **Sender's Password** fields.
5. Click the **Next** button to begin the Central Repository update process.

 **Notes**:

- The same recipients will be used for all selected alerts and reports. If you wish to send different alerts or reports to different recipients, you can update them later using the Alert Center Remote Console.

- The Alert Center Remote Console can be also used to create custom alerts and reports.

CHAPTER 9: PCI, PII and Banking Data Discovery

Overview

Databases are being used as the main storage of confidential data. Yet because of the vast quantities of ever growing data stored in corporate databases, finding confidential data presents a major data security challenge for auditors and internal control departments who are often unfamiliar with the internal design of database application in use. The PCI, PII and Banking Data Discovery tool provided by DB Audit allows them to quickly find and identify databases and database tables containing such confidential data to ensure adequate data protection.

The following common terms are used in this topic to refer to different types of confidential data:

PCI – Payment Card Industry – The industry enforces very strict regulations for data security described in the Payment Card Industry Data Security Standard (**PCI DSS**) https://www.pcisecuritystandards.org./pdfs/pci_dss_v1-1.pdf. This standard has been designed to help protect the integrity of the credit card systems and to help mitigate the risk of fraud and identity theft to credit card holders. The standard is adopted by both VISA and Mastercard and automatically applies to card association members, merchants, and service providers that store, transmit, or process credit card data.

PCI data includes credit card number in conjunction with a individual or corporate name.

PII – Personally Identifying Information – Any information, which can potentially be used to uniquely identify, contact, or locate a single person falls under this category.

The European Union Data Protection Directive was issued in 1995. The directive establishes guidelines that the 25 EU member states must adhere to when monitoring workforce activity and collecting Personally Identifiable Information. As defined in Wikipedia on-line encyclopedia "Although the concept of PII is ancient, it has become much more important as information technology and the Internet have made it easier to collect PII, leading to a profitable market in collecting and reselling PII. PII can also be exploited by criminals to stalk or steal the identity of a person, or to plan a person's murder or robbery, among other crimes. As a response to these threats, many web site privacy policies specifically address the collection of PII, and lawmakers have enacted a series of legislation to limit the distribution and accessibility of PII."

In 2006, USA government issued a Data Security Directive that instructed all federal agencies to comply with specific data security guidelines issued by the National Institute of Standards and Technology (NIST). Compliance with NIST guidelines requires more than network security and use of firewalls; it requires strong data encryption and access authorization as well as a guaranteed fast response if a breach does occur.

PII data includes such pieces of data as person first, last and family name, Social Security numbers, passport numbers, date or place of birth, mother's maiden name, credit card or bank account number, Tax Payer IDs, and anything else that can be used for personal identification.

Banking Information – This type of data is similar to PCI data, but covers somewhat broader range of financial information including, but not limited to bank account numbers, bank routing numbers (ABA), bank SWIFT codes and so on

Many regulations have been developed on all levels to govern the use and protecting of banking data.

How It Works

DB Audit can perform four different types of database searches:

1. Search database catalog tables for any references to database tables and columns with common names used for storing confidential information.
2. Search database tables storing data in an encrypted format.
3. Search data stored in database tables using pre-configured data pattern matching rules for common PCI, PII and banking data formats.
4. Search data stored in database tables using user-provides data patterns

The internal search rules have been designed with the high search performance in mind. Scanning of a multi-database system using first and second search types typically takes less than 10 seconds. The search using data patterns matching is also highly optimized, although the search time can significantly vary on different database systems and greatly depending the database server performance, on the number of business tables in the database, the pattern matching method and some other factors. The internal search queries constructed by DB Audit use selective data analysis and do not need to scan the entire contents of every table. Moreover, for faster performance, the searches automatically bypass all database system tables.

Running PCI, PII and Banking Data Discovery Utility in Interactive Mode

In this mode, you can run the graphical version of the PII, CPI and Banking Data Search utility that allows you to scan individual database servers and review the results immediately, including previews of the found data.

Use the following method to start the search in graphical interactive mode:

1. In DB Audit Management Console main menu, connect to the database server you want to search.
2. Click **Tools/Search for PII, CPI and Banking Data** top-level menu, or alternatively, on the DB Audit Start Page, click the box with **Find Tables with PII, PCI Data** label. This will start the graphical Find Data Wizard.
3. Follow the prompts displayed by the Find Data Wizard. See the Search Options topic for detailed description of the supported search options.

Running PCI, PII and Banking Data Discovery Utility in Non-interactive Mode

You can schedule non-interactive runs of the PII, CPI and Banking Data Search utility. The non-graphical version of this utility is available in DB Audit API and available with Alert Center component.

The version shipped with the Alert Center, automatically scans all database servers registered with the Alert Center and then generates and emails a single "delta" report displaying only the differences between search utility runs for all servers at once. If a new table containing PCI data has been created or populated since the last run on server A and another table with user names and passwords has been created on server B, an automatically generated report will notify you about both changes.

Use the following method to schedule non-interactive PCI/PII and Banking data search using the Alert Center:

1. In DB Audit Management Console main menu, click **Tools > Alter Center** command, or alternatively, on the DB Audit Start Page, click the box with **Find Tables with PII, PCI Data** label. This will start the Alert Center Remote Console.
2. Connect to the Alert Center.
3. Click **Center Audit Repository based Alerts and Reports** top-level item displayed in the alerts and reports browser. Follow the prompts displayed by the Wizard.
4. Click **Alerts > New Report** command in the top-level menu. The Report Configuration Screen will appear.
5. There is nothing to choose on the first configuration screen. Click the **Next** button to advance to the next step.
6. In the **Reports** drop-down list, select **Discover Hidden PII/PCI and Banking Data (All Servers)** item and click the **Next** button again.
7. Choose the desired report schedule and recipient. Click the **Finish** button.

Notes: In non-interactive mode, you cannot pick individual search options for individual servers. The search utility automatically scans all servers registered with the Alert Center using all supported options for each server type with their default settings.

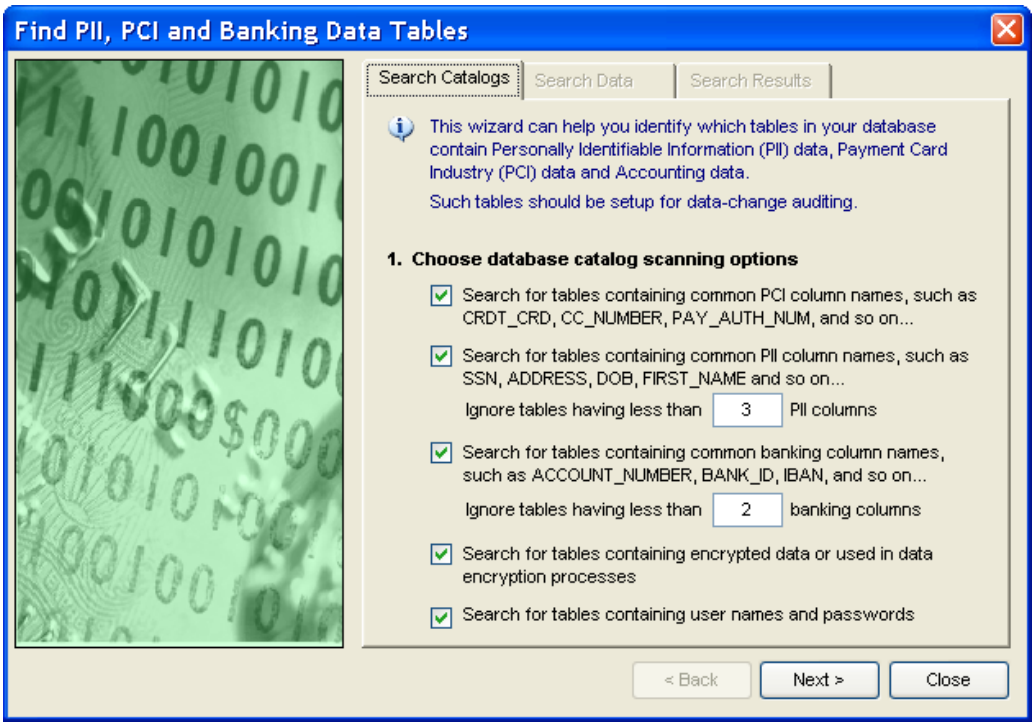
If you require using custom settings for different servers, consider licensing the DB Audit API. The API allows running searches using single-line command line batch files with command line parameters as well as it allows writing sophisticated programs taking full advantage of the programmatic API functions.

Search Options

Database catalog scanning options

Search for tables containing common PCI column names, such as CRDT_CRD, CC_NUMBER, PAY_AUTH_NUM, and so on... This option instructs DB Audit to scan database system catalog tables for references to tables whose column names match names commonly used for storing PCI data. DB Audit uses the internal data dictionary of such common names in English, French and Spanish.

Search for tables containing common PII column names, such as SSN, ADDRESS, DOB, FIRST_NAME and so on... This option instructs DB Audit to scan database system catalog tables for references to tables whose column names match names commonly used for storing PII data. DB Audit uses the internal data dictionary of such common names in English, French and Spanish. The **Ignore tables having less than N columns** option can be used to eliminate false positives so that if, for example, a table contains only first and last name of a person, and other columns not in PII dictionary, such table will not appear in search results. The threshold number can be in 1 - 999 range. If you specify one, any table containing at least one PII data item would be found.



Search for tables containing common banking column names, such as ACCOUNT_NUMBER, BANK_ID, IBAN, and so on... - This option instructs DB Audit to scan database system catalog tables for references to tables whose column names match names commonly used for storing banking data. DB Audit uses the internal data dictionary of such common names in English, French and Spanish. The **Ignore tables having less than N columns** option can be used to eliminate false positives, so that if for example, a table contains only "account" column, and columns not in a banking data dictionary, such table will not appear in search results. The threshold number can be in 1 - 999 range. If you specify 1, any table containing at least one banking data would be found.

Search for tables containing encrypted data or used in data encryption processes - This option instructs DB Audit to scan database system catalog tables and program code for stored procedures and packages containing references to data encryption procedures and their dependent tables. Note that if code of such procedures is not available or stored in the database in some obfuscated or encrypted format, DB Audit might be unable to find the correct references. Cross-database references are not currently supported.

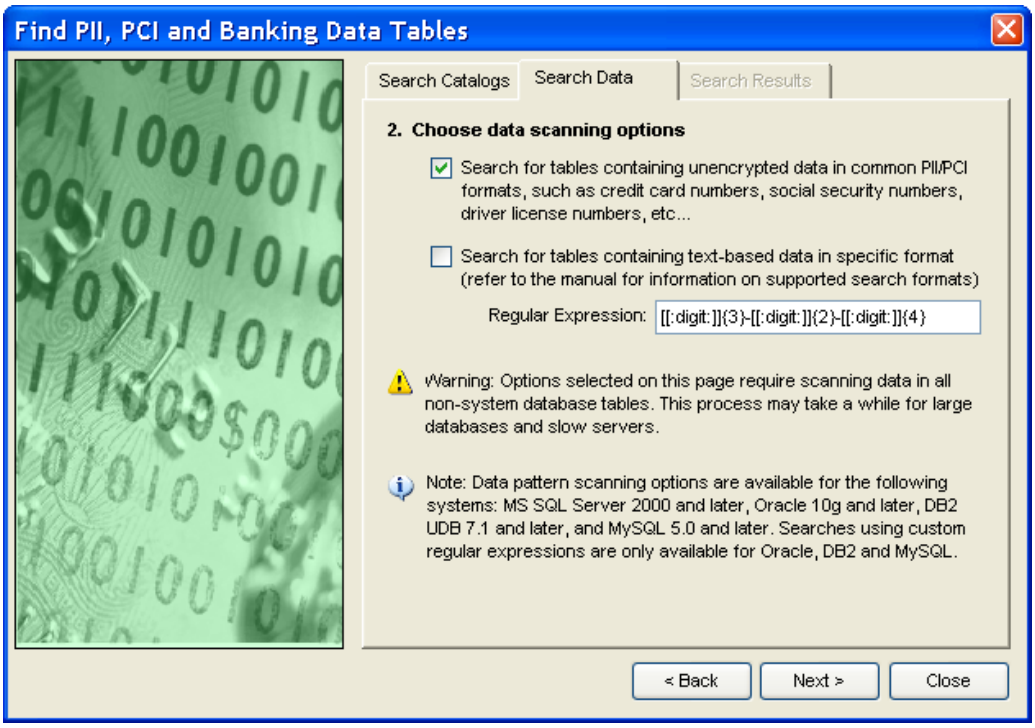
Search for tables containing user names and passwords - This option instructs DB Audit to scan database system catalog tables for references to tables whose column names match names commonly used for storing user names and passwords. DB Audit uses the internal data dictionary of such common names in English, French and Spanish.

Table data scanning options

Search for tables containing unencrypted data in common PII/PCI formats, such as credit card numbers, social security numbers, driver license numbers, etc... This option instructs DB Audit to scan data in database tables to find matches against common formats used for storing SSN numbers, Medicare, Driver License numbers and some other.

DB Audit will use regular expression based searches internally to find confidential data stored in the database tables.

DB2: To run data matching searches, you must install **dbauditRunner.jar** file on your DB2 system. This file provides functions for DB2 system-level auditing, but it also provides support for Regular Expressions. If you are installing this file for the first time, don't forget to update the CLASSPATH environment variable for the DB2 instance owner and to bounce the DB2 server as described in DB2: Enabling system audit topic in CHAPTER 3. **You don't have to enable the system auditing for this function to work. Simply follow the instructions provided for updating the environment variables.** Also, note that your DB2 instance must be configured to use Java **JDK version 1.4 or later** for the data search feature to work properly. If it is configured to use earlier versions of JDK, the data search will fail silently and will be unable to find any qualifying confidential data.



Search for tables containing text-based data in a specific format - this option is similar to the previous option except that you can specify a new custom search pattern based on the regular expression syntax supported by your database system

DB2: The **dbauditRunner.jar** file is also required for this option. See installation instructions described for the previous option.

Using Custom Search Patterns

DB Audit allows you to specify custom search patterns in searches for confidential data in the database. Because the data search queries are executed on the database side, the format of search patterns, also called Regular Expressions, is database type and version dependent. Different database systems support different syntax and extensions.

Oracle, MySQL: Oracle databases version 10g and later and MySQL databases v5.0 and later support almost identical syntax for Regular Expressions. When specifying your custom search criteria,

refer to your database manual for detailed instructions on the supported syntax. The following example demonstrates an Oracle and MySQL compatible Regular Expression that can be used with these database systems for searching Social Security Numbers.

```
[[:digit:]]{3}-[[:digit:]]{2}-[[:digit:]]{4}
```

 **SQL Server:** SQL Server supports a subset of Regular Expressions which, in SQL Server, are called "LIKE patterns." When specifying custom search criteria, refer to your database manual for detailed information about the LIKE syntax. The following example demonstrates a SQL Server LIKE pattern that can be used for searching Social Security Numbers.

```
[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9]
```

 **DB2:** For DB2 databases, DB Audit uses a Java-based UDF function installed in the database. That function is based on the use of Java version 1.4 (or later Java version) implementation of Regular Expressions, which is virtually compatible with the original Perl implementation. To use that function, you must install **dbauditRunner.jar** file on your DB2 system. This file provides functions for DB2 system-level auditing, but it also provides support for Regular Expressions.

If you are installing this file first time, don't forget to update the CLASSPATH environment variable for the DB2 instance owner and to bounce the DB2 server as described in DB2: Enabling system audit topic in CHAPTER 3. **You don't have to enable system auditing for this function to work. Simply follow the instructions provided for updating the environment variables.** Also, note that your DB2 instance must be configured to use Java **JDK version 1.4 or later** for the data search feature to work properly. If it is configured to use earlier versions of JDK, the data search will fail silently and will be unable to find any qualifying confidential data. The following example demonstrates Regular Expression that can be used with DB2 for searching Social Security Numbers.

```
[0-9]{3}\-[0-9]{2}\-[0-9]{4}
```

Searching Arbitrary Types of Encrypted information

The PII, PCI, and Banking Data Search Tools are not limited to specific data formats or information types. If you are already storing some encrypted information in the database, you can use the described tools to quickly find such places and ensure that access to these places is well protected and that adequate auditing controls have been enabled

Displaying, Saving, and Printing Search Results

After you complete the required search options, click the **Next** button on the **Find PII, PCI and Banking data** dialog to start the search process. The search results begin appearing on **Search Results** tab as quickly as the first suspect tables are located on the database server. During an active search operation, the progress bar appears at the bottom of the DB Audit screen indicating the current progress. The disappearance of the progress bar indicates the search process completion. A vertical scroll bar automatically appears on right side of the **Search Results** tab if too many suspect tables

have been found to list them all on a single page.

After the process is complete, you can save or print the results.

To save the report as a text, XML, Excel file, or another supported format, click the **Save** button displayed above the search results. The Save dialog will appear. In the **File Type** drop-down, select the required file type and press **OK.**

To print the displayed report, click the **Print** button displayed above the search results.

What to Do Next

The data in all suspect tables should be reviewed. All essential tables should be set up for data-change auditing in order to maintain an auditable trail of all data changes occurring in these tables.

Tip: For compliance with many regulations, storage of data in encrypted form is insufficient. Continuous data change monitoring and protection are also required. An audit trail must be available for every change in tables containing any type of confidential data.

Refer to CHAPTER 4, Data Change Auditing for instructions on how to set up data auditing.

The following paragraphs describe why it is important to use data-change auditing for tables containing confidential information, and why other auditing methods cannot be used as a substitute:

- System-level auditing can record data change events including information about who, when and how these changes were made and, in many cases, the complete SQL command used to make a changes. Yet, having the text of the command is insufficient in many cases to know the impact of the command, in which case you may be in non-compliance with SOX and many other regulations. Consider the following example,

  ```
  UPDATE hr.employee SET hire_date = DateAdd(mm, -3, hire_date)
  WHERE emp_name LIKE 'A%'
  ```

  This command will change the hire date of certain employees whose names begins with the letter A. Yet if you look at the state of the EMPLOYEE table a month later, you will see a different set of employees and will not be able to figure which of them have been affected by this command.

- Network based firewalls and traffic analysis tools have an even bigger set of problems:

  1) They can capture SQL command text as in the system-audit case example, but only if the command is executed remotely. If the command is executed by a staff member using local database session, that command doesn't generate network traffic and therefore is completely invisible to network monitoring tools.

  2) As explained above, having the text of the command is insufficient in many cases to know the impact of the command, which may put you in a non-compliance situation with SOX and many other regulations.

  3) Many modern applications use various performance optimization methods like server side cursors and variable binding. If you look at the text of executed commands, you may see something like this:

  <begin command 1>

```
EXEC sp_cursoropen @cursor OUTPUT, ' UPDATE hr.employee SET hire_date
= @P1 WHERE emp_name LIKE @P2', 2, 8193
<end command 1>

<begin command 2>
 EXEC sp_cursorexecute @cursor, @P1=@P1, @P=P2
<end command 2>

<begin command 3>
 EXEC sp_cursorexecute @cursor, @P1=@P3, @P=P4
<end command 3>

<begin command 4>
 EXEC sp_cursorclose @cursor
<end command 4>
```

As you can see, there are four separate commands that are executed separately, making two separate updates in the employee table. These updates affect multiple records and none of the executed commands provides any information about which employees were hit by the update.

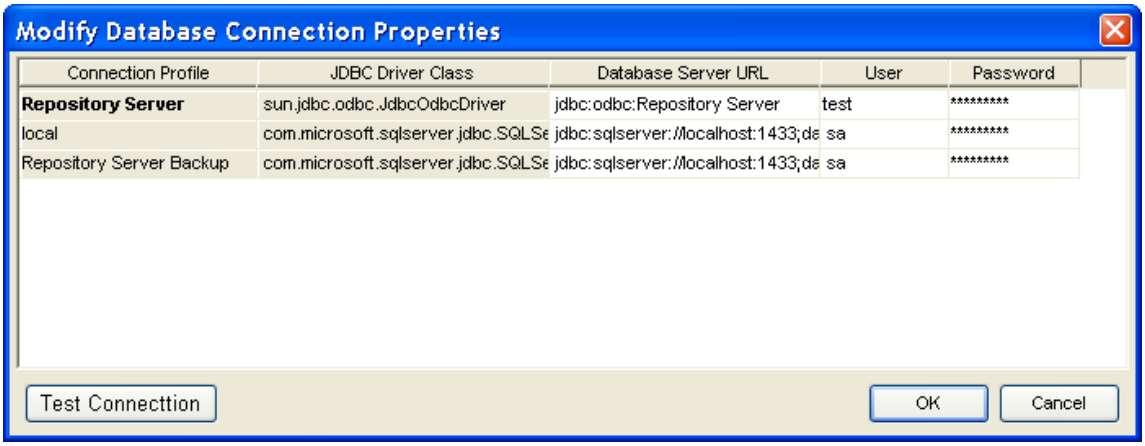
- Database log analysis tools are capable of showing precisely the impact of the data change commands, but their scope is limited to the data available in the current log file, and they are also unable to provide the complete context of the user session that executed updates. If you need to perform a forensic analysis of the data changes involving past activities, the required data will likely gone from the logs and not available for analysis.

Updating Passwords and Connection Settings

Using the Alert Center Remote Console you can easily modify connections settings for database profiles already registered with the Alert Center.

1. Start the Alert Center Remote Console and connect to the Alert Center as described in CHAPTER 6.

2. In the Alert Center Remote Console, click **File > Options** menu. The **Modify Database Connection Properties** dialog will appear.



3. Modify connection strings (Database Server URL) user names and passwords as required. Use the **Test Connection** button to determine whether the settings in the highlighted line have been entered correctly.

   Tip: See Adding, removing, and updating server registration in the central repository topic in this chapter for more information on supported Database Server URL.

4. Click the **OK** button when done.

CHAPTER 10: Security Management

Overview

DB Audit provides a one-stop, multi-database solution for managing database logins, users, security settings and permissions. It provides a unified set of easy-to-use graphical interfaces for managing database users across multiple server types and versions from one central location.

Using DB Audit's security management tools you can

- Control which users and user groups can access your database systems
- Control which items and activities in the database are available to which database users, applications and database user roles
- Identify inappropriate permissions and access levels
- Identity effective security settings and data access paths
- Identify which users and user groups are permitted to access files outside of the database using built-in database file access and extended procedures.

Oracle Database Security Management

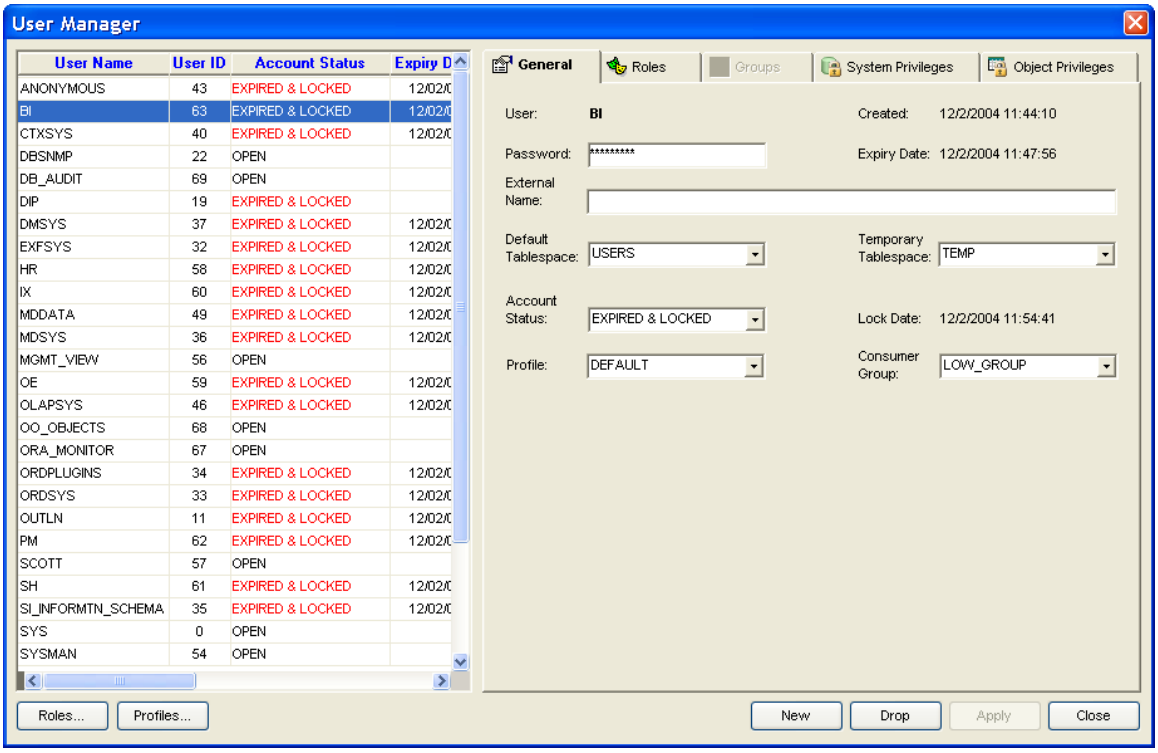
Managing Database Users

To start the **User Manager,** use either of the following methods:

- Click the **Tools > User Management** menu or click the **Manage**  button on the application toolbar. The **User Manager** dialog will appear.
- On the DB Audit Start Page, click the box with **Preventive Security** label.

The **User Manager** dialog consists of two parts: The left side of the screen lists all database users and their properties in a single table-like list. The right hand side is used to display and edit user properties, to assign roles, and to assign various system-level and object-level privileges. The settings and properties displayed on the right side are associated with the user whose name is selected in the user list on the left side of the screen.

You can click on column headers available in the list and on property pages, to rearrange how user names or properties appear in the list.



User Properties

Note: Different Oracle database versions support different sets of user properties. On the **General** tab page, DB Audit displays only options supported in the database you are connected to.

User – Displays the account name of the current Oracle user.

Created – Displays the date and time the user account was created. This property is automatic and cannot be changed.

Password – Specifies the user's password. Use this property when creating Oracle users identified locally by the database; that is, to specify the password the user must enter to log on to the database

External Name – Specifies the External Name when creating an external user. External users must be authenticated by an external service, such as an operating system or a third-party service. In this case, Oracle database relies on the login authentication of the operating system to ensure that a specific operating system user has access to a specific database user.

Important Note: Oracle strongly recommends that you not use external authentication with operating systems that have inherently weak login security. For more information, see Oracle Database Administrator's Guide.

Default Tablespace – Specifies the default tablespace for objects the user creates. If you omit this property, the user's objects are stored in the database default tablespace. If no default tablespace is specified for the database, the user's objects are stored in the SYSTEM tablespace.

Restrictions: You cannot specify a locally managed tablespace, including an undo tablespace, or a dictionary-managed temporary tablespace as a user's default tablespace.

Temporary Tablespace – Specifies the tablespace for the user's temporary segments. If you omit this property, the user's temporary segments are stored in the database default temporary tablespace or, if none has been specified, in the SYSTEM tablespace.

 **Restrictions:** The specified tablespace must be a temporary tablespace and must have a standard block size. The tablespace cannot be an undo tablespace or a tablespace with automatic segment-space management

Account Status – Specifies user's account status; one of the following:

- **OPEN** – user account is active and enabled
- **LOCKED** – user account is inactive and disabled.
- **EXPIRED** – user account is enabled, but the password is expired; user must change his or her password on the next logon to the database
- **EXPIRED & LOCKED** – user account is inactive and disabled; in addition, user password is expired. If the account status later changes to **OPEN,** the password expiry status remains in affect.

Expiry Date – Date and time when the user password expired (for expired accounts) or will expire (for open accounts).

Lock Date – Date and time when the user account was locked. This property is automatic and cannot be changed.

Profile – Specifies the default profile assigned to the user. The profile limits the amount of database resources the user can use. If you omit this property, the Oracle database assigns the DEFAULT profile to the user.

Consumer Group – Specifies the default consumer group assigned to the user. If Oracle's Database Resource Manager is not activated, this property will have no effect on the user.

 **Tip:** See your Oracle documentation for additional information about user account properties and how to use them.

Enabling and Disabling User Accounts

You can use the **General** page to change user account status. Select the LOCKED or EXPIRED & LOCKED status from **Account Status** drop-down list to disable the user account. Select the OPEN status to re-enable the user account. When you click the **Apply** button, all of your changes, including changes on other tabs, are applied at once.

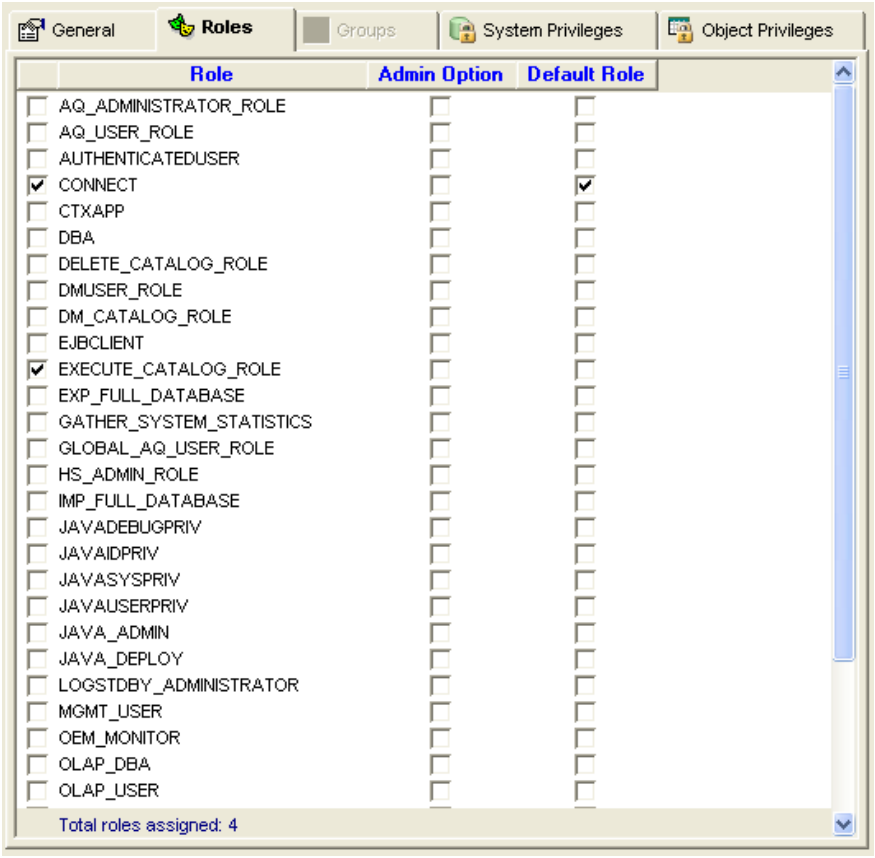
Forcing Users to Change Their Passwords

You can use the **General** page to change user account status. Select the EXPIRED status from **Account Status** drop-down list to expire user password. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

Granting and Revoking Roles

The **Roles** tab on the User Manager dialog can be used to control which roles are assigned to the

selected user.



This is a multiple-choice screen. You can select as many roles as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

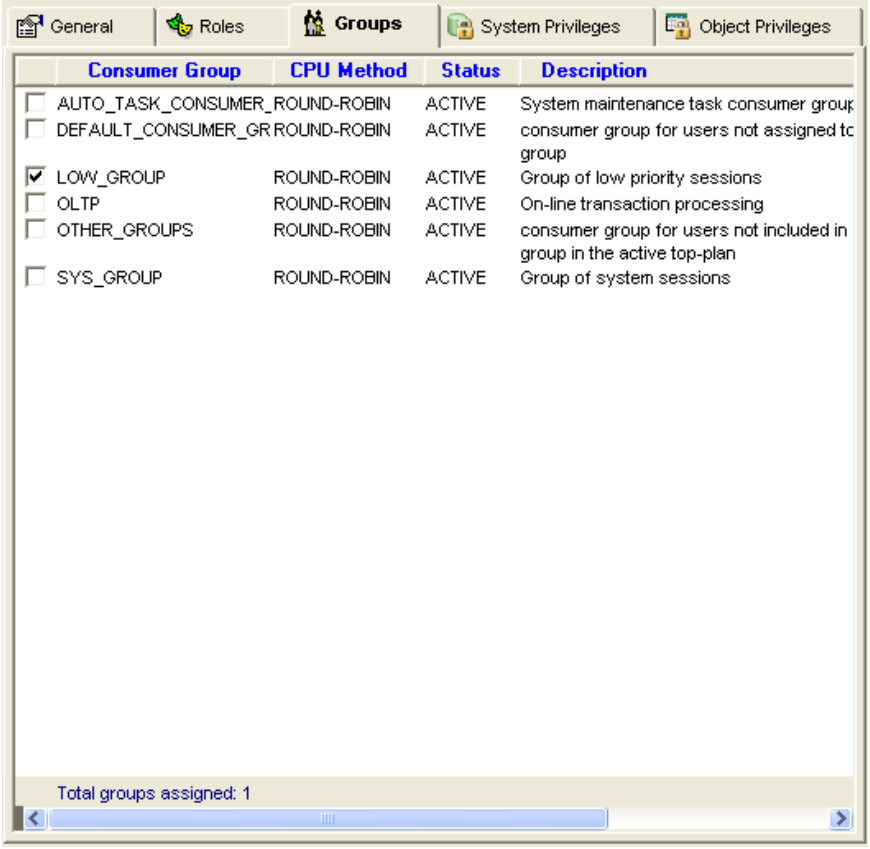
To grant a role to the selected user, place a checkmark in the left-most column. To revoke a role from the selected user, clear the corresponding checkmark.

Checking the **Admin Option** box enables the user to grant this role to someone else. Checking the **Default Role** option instructs Oracle to automatically give the selected user access to the role when a connection is established, even if the role is password-protected and normally requires the user to execute a SET ROLE command to enable role for the current session.

The **total** field shows the number of roles currently selected.

Granting and Revoking Consumer Groups

The **Groups** tab page on the User Manager dialog can be used to control which consumer groups are enabled for the selected user; in other words, which groups are granted to the selected user.



This is a multiple-choice screen. You can select as many groups as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

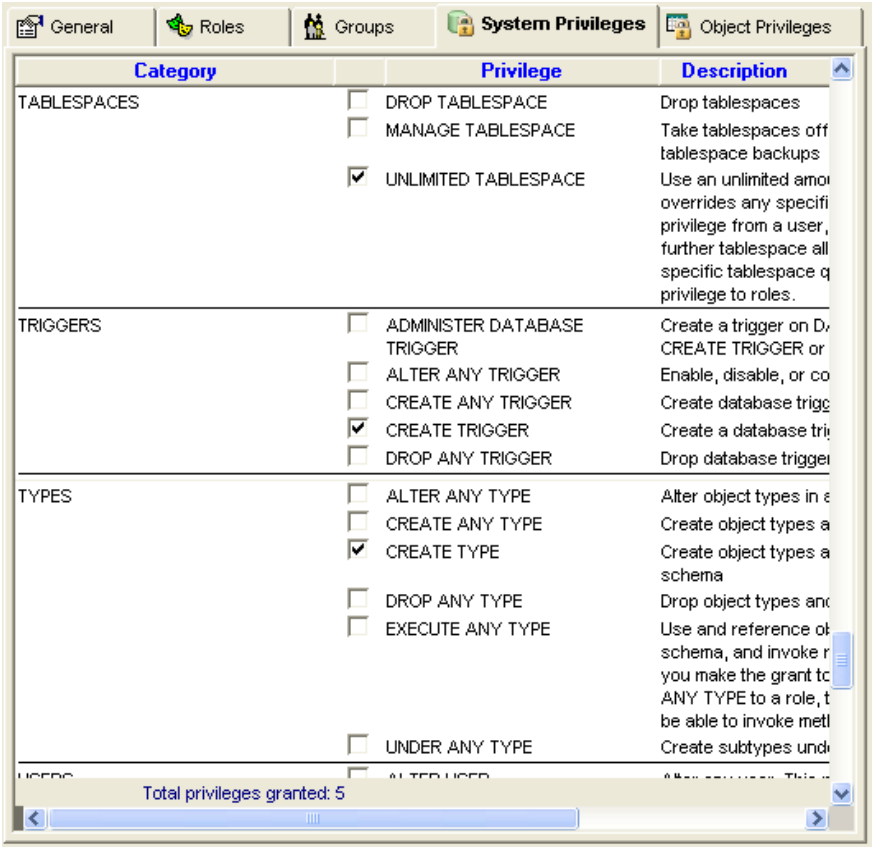
To grant a group to the selected user, place a checkmark in the left-most column. To revoke a group from the selected user, clear the corresponding checkmark.

Tip: Assigning a consumer group to a user is not the same as using it. DB Audit simply grants the user privileges to switch to and use the assigned group whenever that user wants. Keep in mind that a database session can be switched to only one consumer group at a time. Yet, you may have a number of reasons to assign multiple groups to a single user. For example, you may want to allow the user (or a database application run in the context of that user account) to be able to switch to different groups and to use different resource usage plans during different times, for instance, using "LOW" usage plans for daily report processing, while using "HIGH" usage plans for mission critical business processing.

The **total** field shows how many consumer groups are currently selected.

Granting and Revoking System Privileges

The **System Privileges** tab page on the User Manager dialog can be used to control which system privileges are enabled for the selected user; in other words, which privileges are granted to the selected user.



This is a multiple choice screen. You can select as many privileges as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

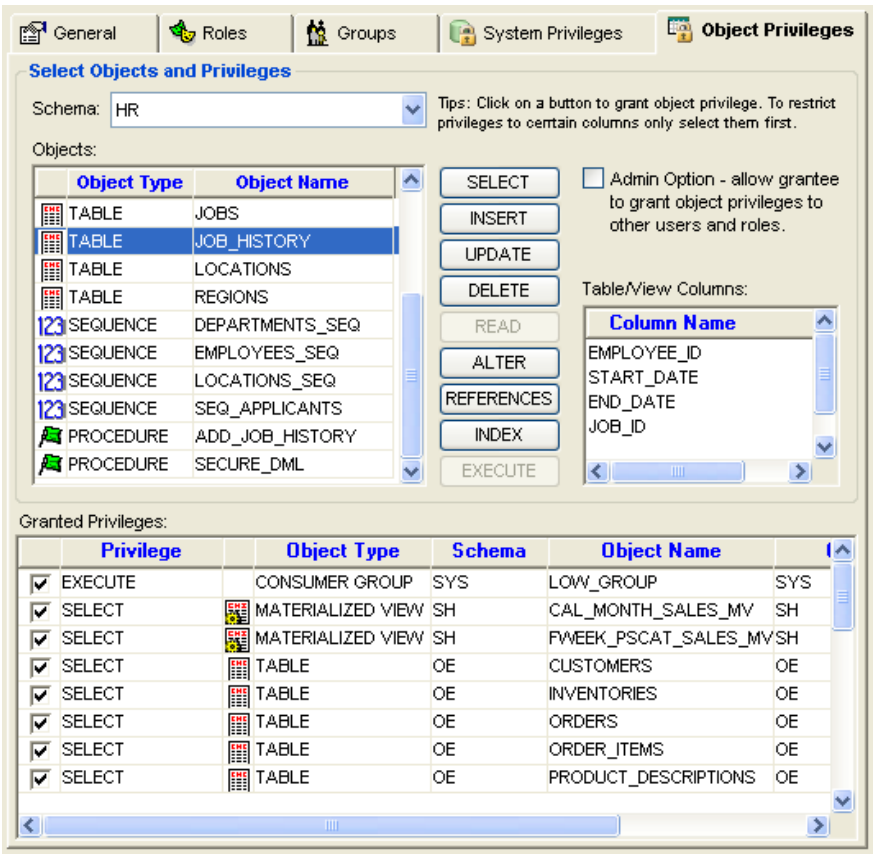
For your convenience all system privileged are grouped and sorted by category and comments explaining each privilege displayed in the right-most column.

To grant a privilege to the selected user, place a checkmark in the left-most column. To revoke a privilege from the selected user, clear the corresponding checkmark.

The **total** field shows how many privileges are selected.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the User Manager dialog can be used to control which database object access privileges are enabled for the selected user; in other words, which privileges are granted to the selected user.



This is a multiple choice screen. You can select as many privileges as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

The **Object Privileges** tab page is split into two logically separate parts. The top part, **Select Objects and Privileges,** is used to grant new privileges. The bottom part, **Granted Privileges,** displays all object-level privileges that have already been granted to the selected user. It can also be used to revoke these privileges.

To grant a privilege to the selected user:

1. Select object's schema name in the Schema drop-down list. DB Audit will populate the **Objects** list will all grantable objects in specified schema name.
2. Select the object you want to grant to the user in the **Objects** list. If you select a table or view, DB Audit will populate the **Table/View Columns** list displayed on the right hand side with names of all columns available in the selected object. It will also enable or disable various "grant" buttons displayed in the middle of the screen according to which privileges can be granted for the selected object type.
3. If necessary, check the **Admin Option** box. Checking the **Admin Option** box enables the user to grant the given privilege to someone else.
4. Click the appropriate "grant" buttons (SELECT, INSERT, and so on) to grant specific privileges to the user. These privileges will be automatically appended to the end of the **Granted Privileges** grid. If you make a mistake, simply uncheck the left-most column in the grid to remove the privilege.

   For UPDATE, INSERT and REFERENCES operations, if required, you can restrict user's

access to specific table or view columns only. Select all required columns in the column list, and then click the UPDATE or INSERT or REFERENCES button once. To select multiple columns, hold down the CTRL key while clicking on column names. DB Audit grants all-columns access to the user when no columns are selected in the column list.

To revoke a privilege from the selected user:

To revoke a privilege from the selected user, clear the corresponding checkmark on the **Granted Privileges** grid.

Creating New Database Users

1. Open **User Manager** dialog using **Tools > Manage Database Users** menu.
2. Click the **New** button.
3. Fill in new user properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages read previous topics in this chapter.
5. Click the **Apply** button to save changes.

Deleting Database Users

1. Open the **User Manager** dialog.
2. On the left side of the dialog select name of the user you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the **YES** button if you are sure that the right user is chosen for deletion.

 **Important Note:** DB Audit automatically deletes all user objects associated with the user schema, including all referential constraints referring to the affected objects from other schemas

Managing User Profiles

To start the **Profile Manager**, click the **Profiles** button on the Oracle User Manager screen. The **Profile Manager** dialog will appear.

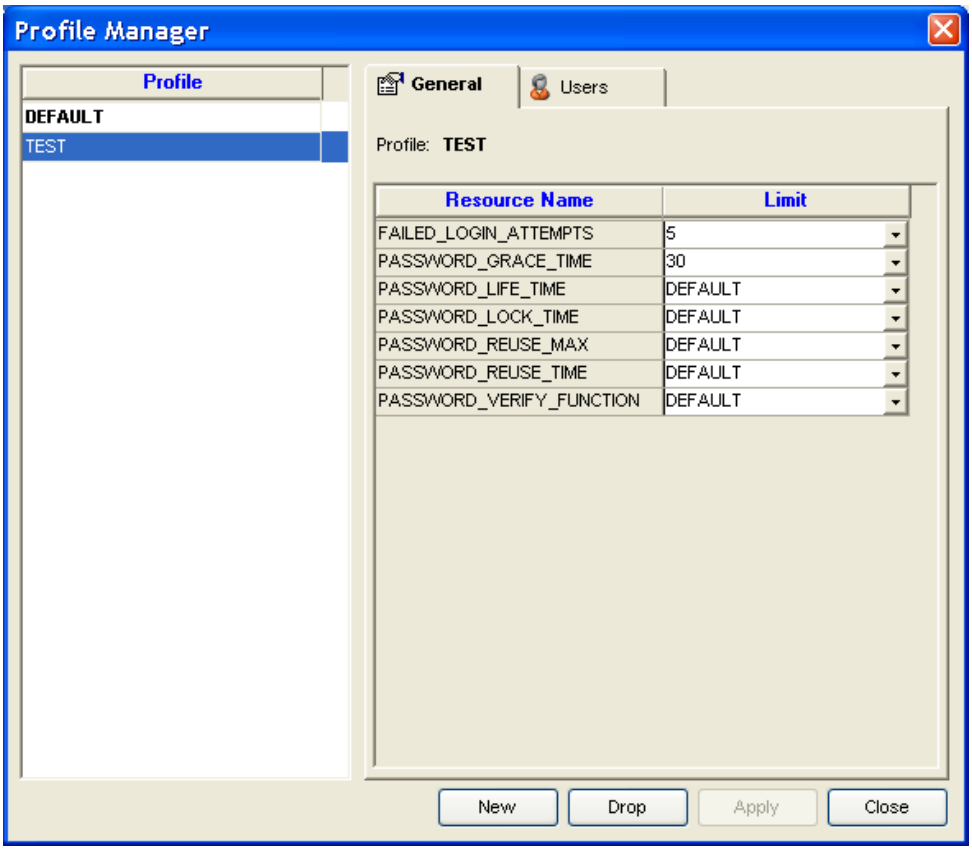
 **Note:** Different Oracle database versions support different sets of profile properties. On the **General** tab page, DB Audit displays only options supported in the database you are connected to.

The **Profile Manager** dialog consists of two parts: The list box on the left hand side lists all database profiles. The right hand side of the dialog screen is used to display and edit profile properties and also is used to assign profiles to users. The settings and properties displayed on the right hand side are associated with the profile whose name is selected in the profile list on the left hand side.

You can click on column headers available in the list and on property pages to rearrange how user

names or properties appear in the list.

Consult your Oracle documentation for detailed information about supported profile properties and their values.



Profile Properties

Database Security

FAILED_LOGIN_ATTEMPTS - Specifies the number of failed logon attempts allowed before the user account is locked out.

Password Security

Parameters that set lengths of time are interpreted in number of days. For testing purposes, you can specify minutes or even seconds as a decimal number. For example, to specify five minutes, enter 0.003472 (this value was calculated as 1/24/60*5).

PASSWORD_LIFE_TIME- Specifies the number of days the same password can be used for authentication. If you also set a value for PASSWORD_GRACE_TIME, the password expires if it is not changed within the additional grace period. When the grace period expires, further connections are rejected. If you do not set a value for PASSWORD_GRACE_TIME, its default of UNLIMITED will cause the database to issue a warning but let the user continue to connect indefinitely.

PASSWORD_REUSE_TIME and **PASSWORD_REUSE_MAX** - These two parameters must be set in conjunction with each other. PASSWORD_REUSE_TIME specifies the number of

days before which a password cannot be reused. PASSWORD_REUSE_MAX specifies the number of password changes required before the current password can be reused. For these parameters to have an effect, you must specify an integer for both of them.

If you specify an integer for both of these parameters, the user cannot reuse a password until the password has been changed x number of times, where x is the value specified on the PASSWORD_REUSE_MAX parameter. Once the password has been changed x number of times, the original password still may not be reused until y number of days have passed, where y is the value specified on the PASSWORD_REUSE_TIME parameter.

For example, if you specify PASSWORD_REUSE_TIME to 30 and PASSWORD_REUSE_MAX to 10, then the user can reuse the password after 30 days if the password has already been changed 10 times.

If you specify an integer for either of these parameters and specify UNLIMITED for the other, then the user can never reuse a password.

If you specify DEFAULT for either parameter, the Oracle Database uses the value defined in the DEFAULT profile. By default, all parameters are set to UNLIMITED in the DEFAULT profile. If you have not changed the default setting of UNLIMITED in the DEFAULT profile, then the database treats the value for that parameter as UNLIMITED.

If you set both of these parameters to UNLIMITED, then the database ignores both of them.

PASSWORD_LOCK_TIME - Specifies the number of days an account will be locked out after the specified number of consecutive failed login attempts.

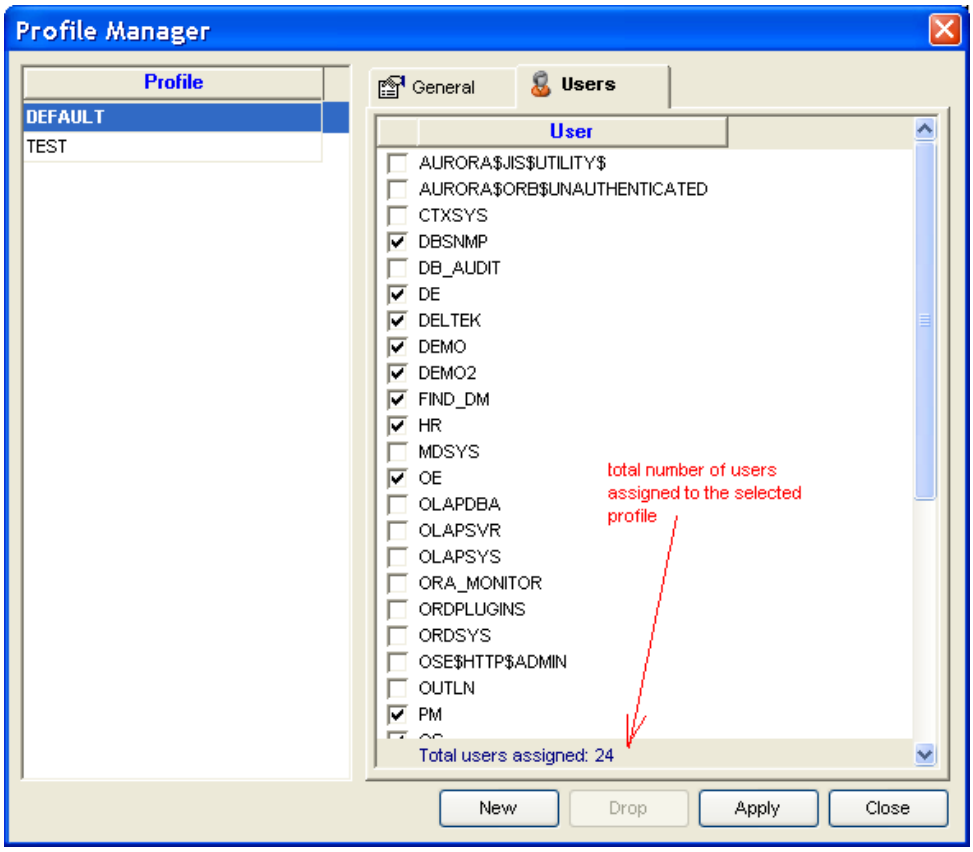
PASSWORD_GRACE_TIME - Specifies the number of days after the grace period begins during which a warning is issued and login is allowed. If the password is not changed during the grace period, the password expires.

PASSWORD_VERIFY_FUNCTION - Specifies the name of the password complexity verification routine. The Oracle database provides a default script, but you can create your own routine to use in its place. Specify NULL to indicate that no password verification is performed.

 **Tip:** PASSWORD_VERIFY_FUNCTION is only **used to verify password complexity** at the time of the user account creation or password change. It is **NOT USED for the user authentication** at the time of user logon to the database.

Altering Profiles

1. Start the **Profile Manager**.
2. Select the required profile in the profile list.
3. On the **General** tab page modify profile properties as needed
4. On the **Users** tab page modify user assignments as needed.



This is a multiple choice screen. You can select as many users as you want. When you click the **Apply** button, all of your changes including all changes on the **General** tab are applied at once.

5. To assign a user to the selected profile place a checkmark in the left-most column. To undo the assignment clear the corresponding checkmark.

   Note: A user is automatically reverted to the DEFAULT profile if the previous profile assignment is cleared.

6. Click the **Apply** button to save changes.

Deleting Profiles

1. Start the **Profile Manager**.
2. In the profile list select the name of the profile you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the **YES** button if you are sure that the right profile is chosen for deletion.

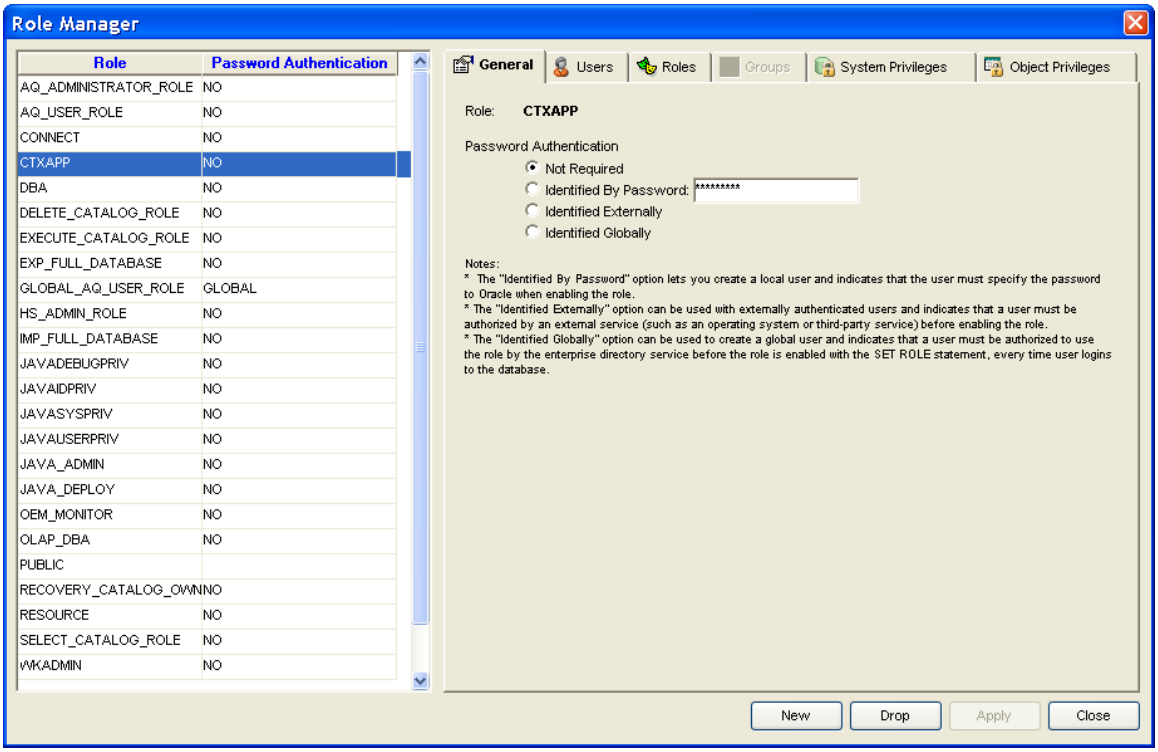
   Important Note: **The DEFAULT profile may not be deleted!**

Creating New Profiles

1. Start the **Profile Manager**.
2. Click the **Add** button
3. On the **General** tab page, modify profile properties as needed.
4. On the **Users** tab page, assign users to the new profile as needed.
5. Click the **Apply** button to save changes.

Managing Database Roles

To start the **Role Manager**, click **Roles** button on the User Manager screen menu. The **Role Manager** dialog will appear.



The **Role Manager** dialog consists of two parts: The left hand side is used to list all database roles and their properties in a single table-like list. The right hand side is used to display and edit role properties, to assign roles to users, and to assign various system-level and object-level privileges to roles. The settings and properties displayed on the right hand side are associated with the role selected in the role list on the left hand side.

You can click on column headers available in the list and on property pages, to rearrange how user names or properties appear in the list.

Consult your Oracle documentation for detailed information about supported role properties and their values.

Role Properties

Different Oracle database versions support different sets of role properties. On the **General** tab page, DB Audit displays only options supported in the database you are connected to.

Role – Displays the selected role name.

Password Authentication – Use these options to specify role password authentication.

Not Required – Indicates that this role is authorized by the database and that no password is required to enable the role.

Identified by Password – Indicates that this a local role specifies the password used to enable the role. The password may contain only single-byte characters from your database character set, regardless of whether the character set also contains multi-byte characters.

Identified Externally – Indicates that the user must be authenticated by an external service and must use the SET ROLE statement to enable this role.

Identified Globally – Indicates that the user must be authenticated by a global enterprise directory service and must use the SET ROLE statement to enable this role.

See your Oracle documentation for more information about role properties and how to use them.

Granting and Revoking Roles to Users

The **Users** tab page on the Role Manager dialog can be used to control which users have been granted the selected role. This page is similar in functionality to the Granting and Revoking Roles page on the User Manager dialog, it just shows the opposite view of "Users granted the selected role" vs. "Roles granted to the selected user." Read Granting and Revoking Roles topic for instructions on how to use this screen.

Granting and Revoking Roles to Roles

The **Roles** tab page on the Role Manager dialog can be used to control which roles in effect have been granted the selected role. This page is similar in functionality to the Granting and Revoking Roles page on the User Manager dialog. Read Granting and Revoking Roles topic for instructions on how to use this screen.

Granting and Revoking Consumer Groups

The **Groups** tab page on the Role Manager dialog can be used to control which consumer groups are enabled for the selected role; in other words, which groups are granted to the selected role.

This page is similar in functionality to the Granting and Revoking Consumer Groups page on the User Manager dialog." Read Granting and Revoking Consumer Groups topic for instructions on how to use this screen.

 **Tips:** Assigning a consumer group to a role is not the same as using it. It simply grants a privilege to switch to and use the assigned group whenever a user wants. Keep in mind that every database session can be switched to only one consumer group at a time. Yet there exist multiple reasons for why you may want to assign multiple groups to a single user. For example, you may want to allow the user (or database applications running under that user account) to switch to different groups and to use different resource usage plans during different times, such as using "LOW" usage plans for daily report processing and using "HIGH" usage plans for other mission critical business processing.

Granting and Revoking System Privileges

The **System Privileges** tab page on the Role Manager dialog can be used to control which system privileges are enabled for the selected role; in other words, which privileges are granted to the selected role.

This page is similar in functionality to the Granting and Revoking System Privileges page on the User Manager dialog." Read Granting and Revoking System Privileges topic for instructions on how to use this screen.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the Role Manager dialog can be used to control which object-level privileges are enabled for the selected role; in other words, which privileges are granted to the selected role.

This page is similar in functionality to the Granting and Revoking Object Privileges page on the User Manager dialog." Read Granting and Revoking Object Privileges topic for instructions on how to use this screen.

Creating New Database Roles

1. Open **Role Manager** dialog.
2. Click the **New** button.
3. Enter new role properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages, read previous topics in this chapter.
5. Click the **Apply** button to save changes.

Deleting Database Roles

1. Open **Role Manager** dialog.
2. On the left hand side of the dialog, select name of the role you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the **YES** button if you are sure that the right role is chosen for deletion.

Important Note: All database users previously assigned to this role will automatically lose all privileges associated with this role.

SQL Server Database Security Management

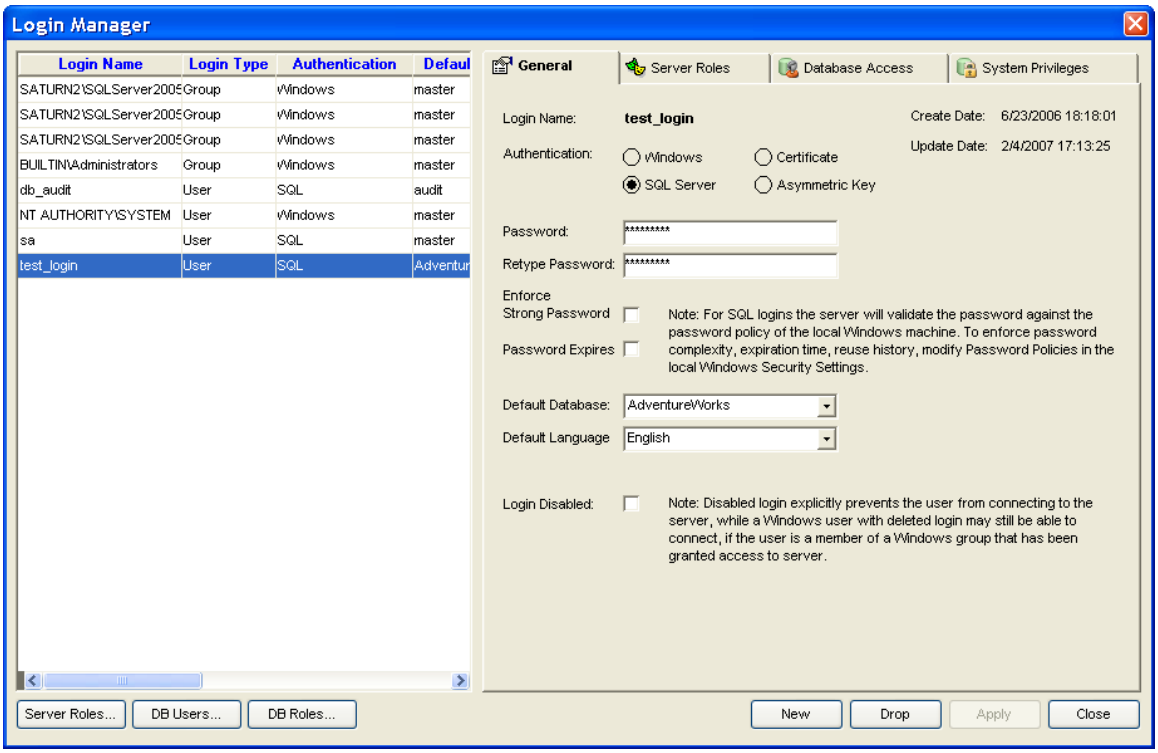
Managing Server Logins

To start the **Login Manager,** use either of the following methods:

- Click **Tools > User Management** menu or click the **Manage** button on the application toolbar. The User Manager dialog will appear.
- On the DB Audit Start Page, click the box with **Preventive Security** label.

The **Login Manager** dialog consists of 2two parts: The left hand side is used to list all database logins and their properties in a single table-like list. The right hand side is used to display and edit login properties associated with the login selected in the login list on the left hand side of the screen.

You can click on column headers available in the list and on property pages to rearrange how user names or properties appear in the list.



Login Properties

Note: Different SQL Server database versions support different sets of login properties. On the **General** tab page, DB Audit displays only options supported in the database system you are connected to.

Login Name – Displays the name of the SQL user account.

Login Type – The type of login such as SQL Server user, Windows User or Windows User Group.

Authentication - Specifies the type of login authentication. Possible types of authentication are Windows, SQL Server, Certificate and Asymmetric Key.

Create Date – Displays the date and time the user account was created. This property is automatic and cannot be changed.

Update Date – Displays the date and time the user account was last modified. This property is automatic and cannot be changed.

Password – Specifies the login password that must be entered by SQL Server users identified locally by the database. This property is available for SQL Server login types.

Enforce Strong Passwords – Specifies whether or not to enforce strong passwords.

> **Important Note:** For SQL logins, the server will validate the password against the password policy of the **local Windows** machine. To enforce password complexity, expiration time and reuse history, you must modify Password Policies in the **local Windows Security Settings**.

Password Expires – Specify whether or not the password has an expiration date.

> **Important Note:** For SQL logins, the password expiration period and reuse times are controlled by security policy of the **local Windows** machine. To set up password complexity, expiration time and password reuse history you must modify Password Policies in the **local Windows Security Settings**.

Default Database– Specify the default database to be assigned to the login. If no default database is specified, the default database is set to MASTER.

> **Restrictions:** The login must be mapped to a valid user in the specified database. See the following “Database Use Management” topic for details on where and how to change database user mapping.

Default Language - Specifies the default language to be assigned to the login. If no default language is specified, the default language is set to the current default language of the server. If the default language of the server is changed in the future, the default language of the login remains .

Login Disabled – Specifies the login status, whether enabled or disabled.

> **Important Notes:** Disabled login explicitly prevents the user from connecting to the SQL Server. In comparison, a Windows user whose login has been deleted may still be able to connect if that user is a member of a Windows user group that has been granted access to the SQL Server.
>
> For SQL Server 2000, DB Audit uses the DENY LOGIN method to effectively disable a user's access to the server. For SQL Server 2005 and 2008, DB Audit may use either the "disabled login" property or the DENY LOGIN method to effectively disable login's access to the server

Tip: See your SQL Server documentation for additional information about logins properties and how to use them.

Enabling and Disabling Logins

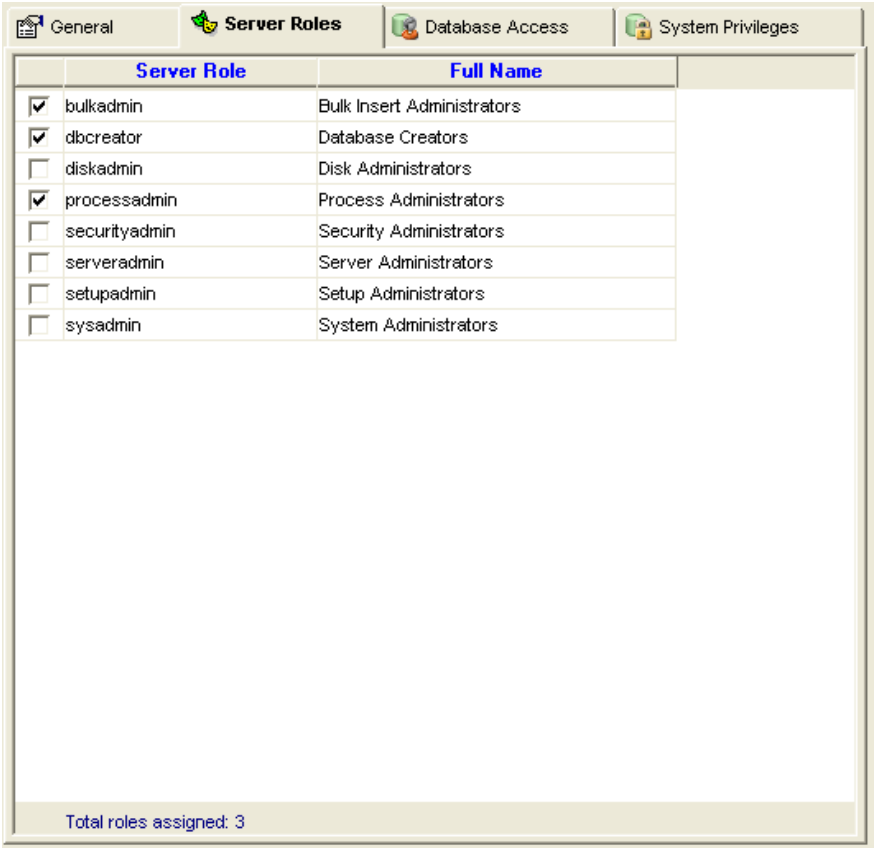
You can use the **General** page to change login status. Check or uncheck the **Login Disabled** checkbox. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

Forcing Users to Change Their Passwords

You can use the **General** page to change login Password Expiration policy. Check or uncheck **Password Expires** checkbox. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

Granting and Revoking Server Roles

The **Roles** tab on the Login Manager dialog can be used to control which server-wide roles are assigned to the selected login.



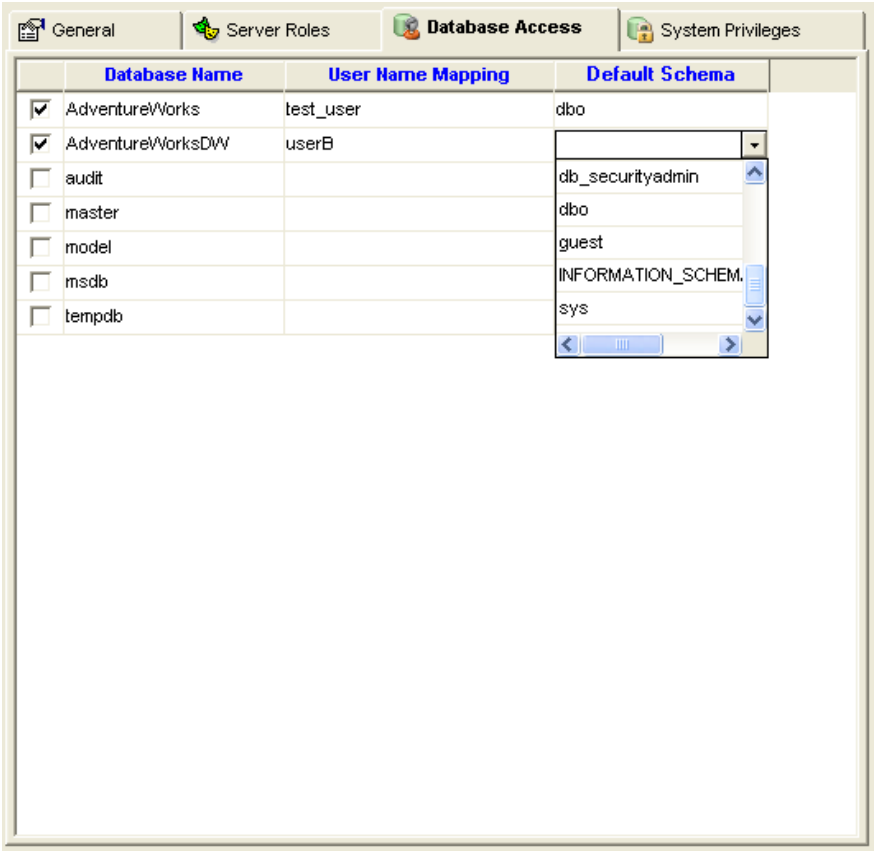
This is a multiple choice screen. You can select as many roles as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

To grant a role to the selected login, place a checkmark in the left-most column. To revoke a role from the selected login, clear the corresponding checkmark.

The **total** field shows how many roles are currently selected.

Granting and Revoking Database Access

The **Database Access** tab page on the Login Manager dialog can be used to control which databases are accessible to the selected login.



This is a multiple choice screen. You can select as many databases as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

To grant access to a database for the selected login, place a checkmark in the left-most column. Enter the desired user name in the target database and choose the database schema to which that user is mapped.

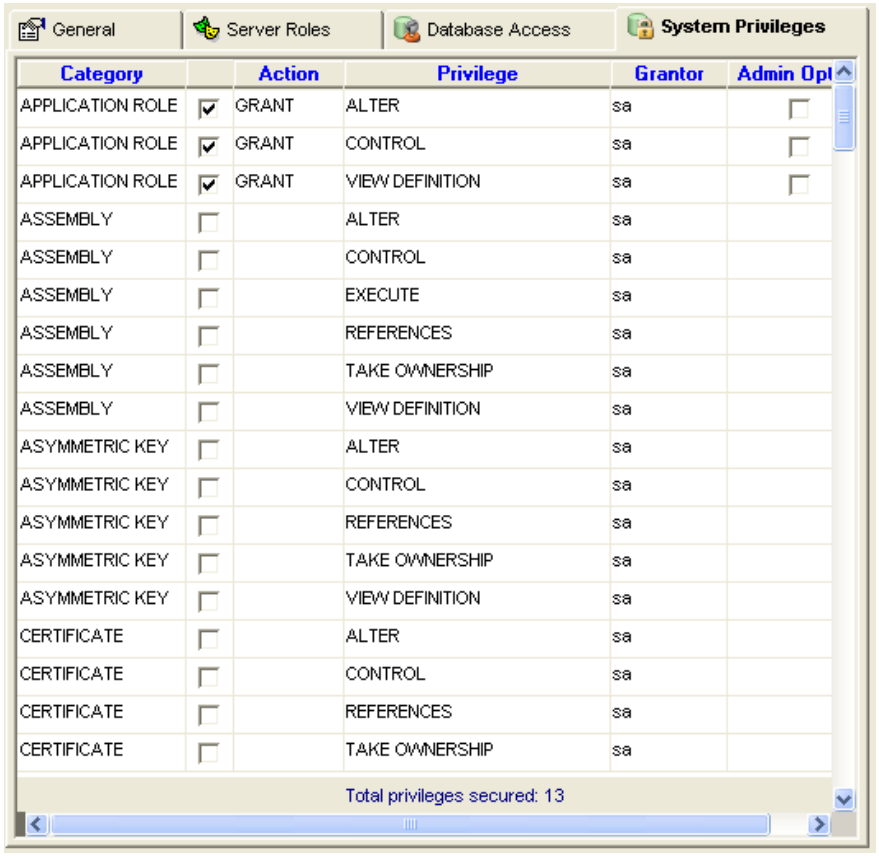
To revoke access for the selected user, clear the corresponding checkmark.

Important Note: You cannot map multiple logins to the database user.

The **total** field shows how many databases are currently selected.

Granting and Revoking System Privileges

The **System Privileges** tab page on the Login Manager dialog can be used to control which system privileges are enabled for the selected login; in other words, which privileges are granted to the selected login. The available privileges are SQL Server version dependent.



This is a multiple choice screen. You can select as many privileges as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

For your convenience all system privileged are grouped and sorted by category.

To grant a privilege to the selected login, place a checkmark in the left-most column. To revoke a privilege from the selected login, clear the corresponding checkmark.

The **total** field shows how many privileges are selected.

Creating New SQL Server Logins

1. Open **Login Manager** dialog using **Tools > Manage Database Users** menu.
2. Click the **New** button.
3. Enter the new login properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages, read previous topics in this chapter.

5. Click the **Apply** button to save changes.

Deleting SQL Server Logins

1. Open the **Login Manager** dialog.
2. On the left hand side of the dialog, select name of the login you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the **YES** button if you are sure that the right login is chosen for deletion.

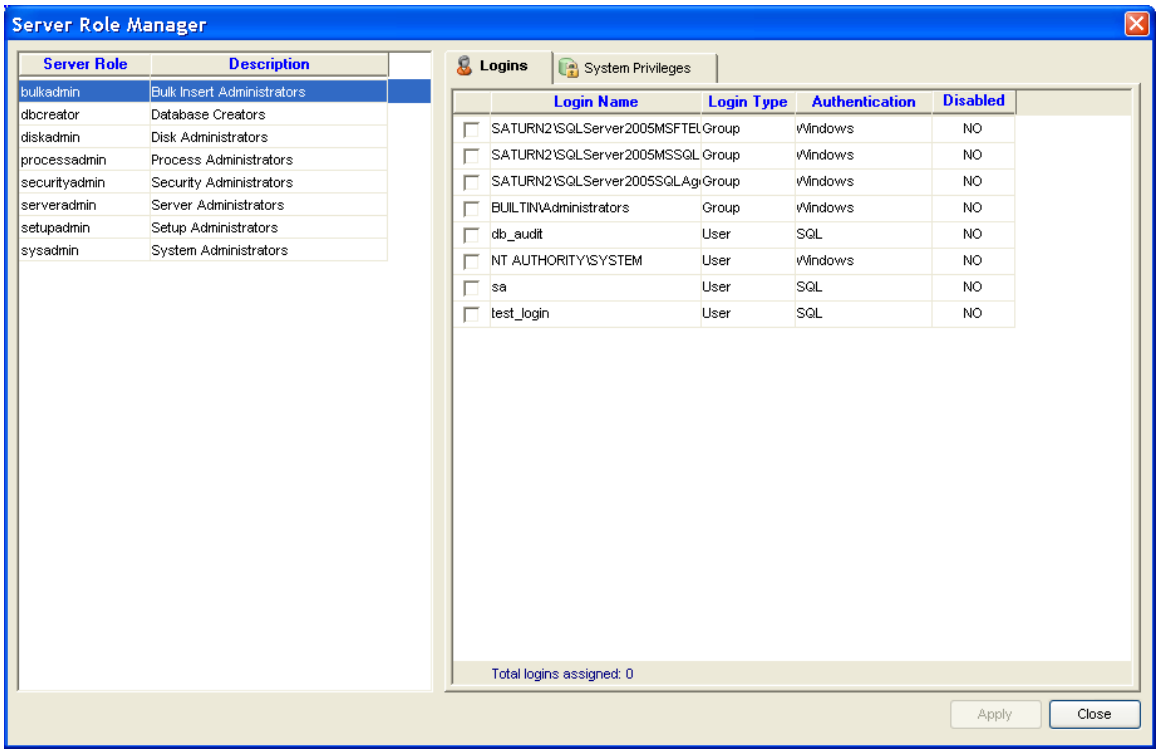
 **Important Note:** DB Audit automatically performs the following additional operations when a login is deleted:

- Deletes all database access and user mappings for the login being deleted.
- Fixes database owners if a login being deleted is set as a database owner for one or more databases.
- Removes login/user aliases for the login being deleted.

Managing Server Roles

To start the **Server Role Manager**, click **Roles** button on the Login Manager screen.

The **Server Role Manager** dialog consists of two parts: The left hand side is used to list all server users and their properties in a single table-like list for all databases available on the server. The right hand side is used to display and edit role properties assigned to the selected server role.



Server Role Properties

Note: Server Role properties are fixed and cannot be modified.

Managing Server Role Associations

Using the **Logins** tab, you can control which logins have been granted the selected role. This is just another view of the login/server-wide role associations. On this screen you can find all logins for a given role while on the Login Manager screen you can find all roles granted to a given login.

The behavior and usage of this screen is similar to the behavior of the Login Manager.

Reviewing Server Role Privileges

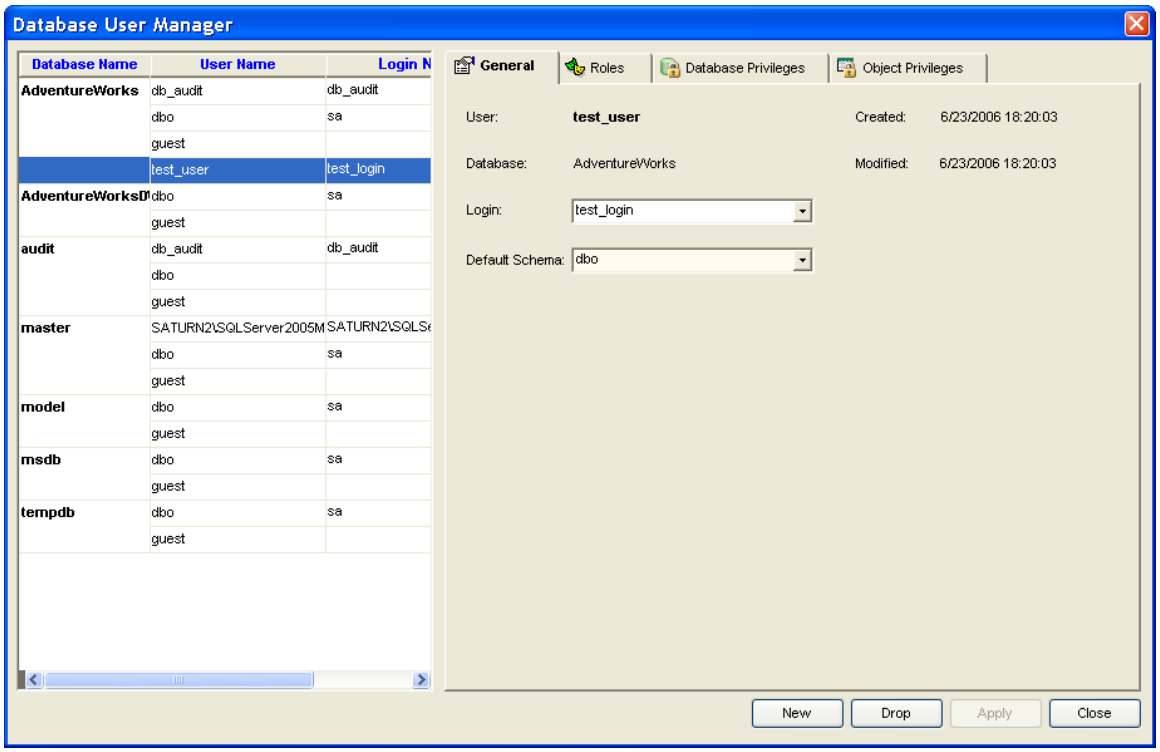
Note: Server Role privileges are fixed and cannot be modified.

To review available privileges for a given role, activate the **System Privileges** tab and then click on the required role name.

Managing Database Users

To start the **Database User Manager**, click **DB Users** button on the Login Manager screen.

The **User Manager** dialog consists of two parts: The left hand side is used to list all database users and their properties in a single table-like list for all databases available on the server. The right hand side is used to display and edit user properties assigned to the selected user.



Database User Properties

Note: Different SQL Server database versions support different sets of user properties. On the **General** tab page, DB Audit displays only options supported in the database system you are connected to.

User – Displays the database user name.

Login – Specifies the login mapped to the selected user.

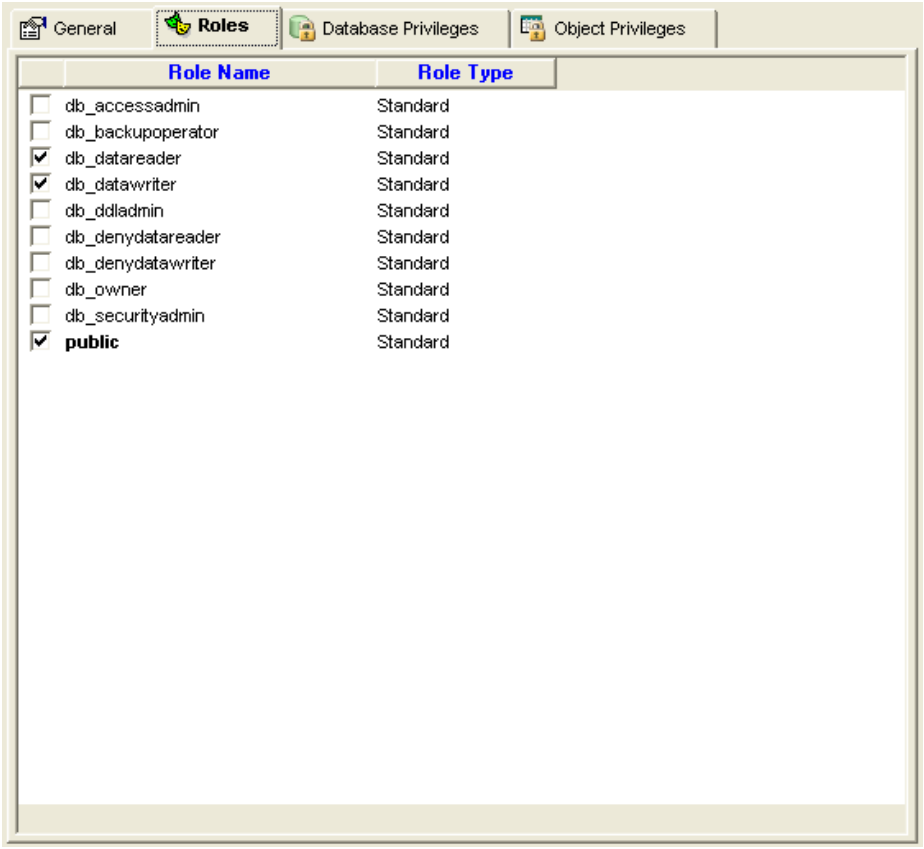
Default Schema - Specifies the default schema associated with the selected database user.

Created – Displays the date and time the user account was created. This property is automatic and cannot be changed.

Updated – Displays the date and time the user account was last modified last. This property is automatic and cannot be changed.

Granting and Revoking Database and Application Roles

The **Roles** tab on the Database User Manager dialog can be used to control which roles are assigned to the selected user.



This is a multiple choice screen. You can select as many roles as you want. When you click the **Apply** button, all of your changes including changes on other tabs are applied at once.

To grant a role to the selected user, place a checkmark in the left-most column. To revoke a role for the selected user, clear the corresponding checkmark.

Note: The value in the **Role Type** column indicates types of the roles. Available options are:

- **Standard –** this represents is a regular database role
- **Application –** this represents an application role

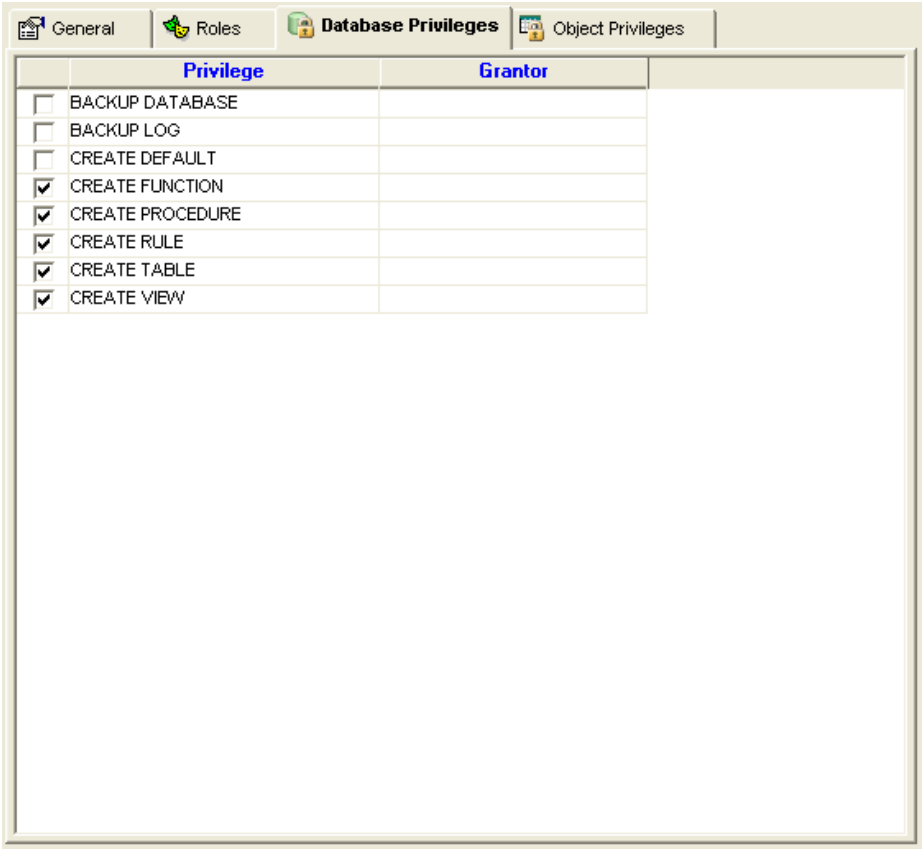
Roles can be created and modified using the Role Manager screen

Restrictions: The default **public** role is available in every database. This role may not be revoked from a database user.

The **total** field shows how many roles are currently selected.

Granting and Revoking Database Privileges

The **Database Privileges** tab page on the Database User Manager dialog can be used to control which database-wide privileges are enabled for the selected user; in other words, which database-wide privileges are granted to the selected user.



This is a multiple choice screen. You can select as many privileges as you want. When you click the **Apply** button, all of your changes, including changes on other tabs are applied at once.

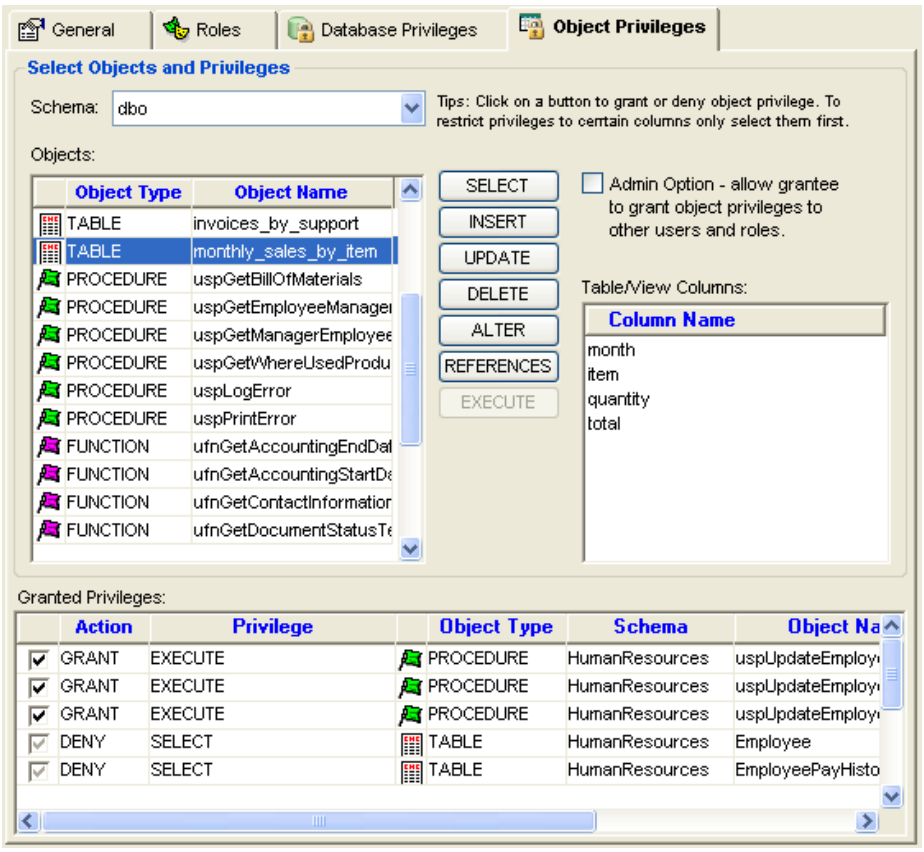
To grant a privilege to the selected user place a checkmark in the left-most column. To revoke a privilege for the selected user, clear the corresponding checkmark.

The **total** field shows how many privileges are currently selected.

Important Note: Server-wide privileges generally supersede database-wide privileges. For example, if a login is granted System Administrator role for the entire server, no matter what privileges you grant tor revoke for the mapped database user accounts, the login can still do pretty much anything in any database on the target server.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the Database User Manager dialog can be used to control which database object access privileges are enabled for the selected user; in other words, which privileges are granted to the selected user.



This is a multiple choice screen. You can select as many privileges as you want. When you click the **Apply** button, all of your changes, including changes on other tabs are applied at once.

The **Object Privileges** tab page is split into two parts. The top part, **Select Objects and Privileges,** is used to grant new privileges. The bottom part, **Granted Privileges,** displays all object-level privileges that have been already granted or denied to the selected user. The bottom part of the screen can also be used to revoke privileges simply by deselecting the appropriate checkmarks in the first column.

To grant a privilege to the selected user:

1. Select object's schema name in the Schema drop-down list. DB Audit will populate the **Objects** list with all grantable objects in specified schema name.
2. Select the object you want to grant to the user in the **Objects** list. If you select a table or view, DB Audit will populate the **Table/View Columns** list displayed on the right hand side with names of all columns available in the selected object. It will also enabled/disable the "grant" buttons displayed in the middle of the screen according to what privileges can be granted for the selected object type.
3. If necessary, check the **Admin Option** box. Checking the **Admin Option** box enables the user to grant the given privilege to someone else.
4. Click the appropriate "grant" buttons (SELECT, INSERT, and so on) to grant specific privileges to the user. These privileges will be automatically appended to the end of the **Granted Privileges** grid. If you make a mistake, simply uncheck the left-most column in the grid to remove the privilege.

For UPDATE, INSERT and REFERENCES operations, you can restrict user's access to specific table or view columns. Select all required columns in the column list, and then click the UPDATE or INSERT or REFERENCES button once. To select multiple columns, hold down the CTRL key while clicking column names. DB Audit grants all-columns access to the user when no columns are selected in the column list.

5. Click the **Action** column in the **Granted Privileges** section and use the drop-down list to GRANT, DENY or REVOKE the selected privilege.
6. Click the **Apply** button

To revoke a privilege from the selected user:

1. To revoke a privilege from the selected user, clear the corresponding checkmark in the **Granted Privileges** grid or click the **Action** column and select REVOKE option from the drop-down list.
2. Click the **Apply** button

Creating New Database Users

1. Open the **Database User Manager** dialog using **DB Users** button on the Login Manager screen.
2. Click the **New** button.
3. Fill in new user properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages, read previous topics in this chapter.
5. Click the **Apply** button to save changes.

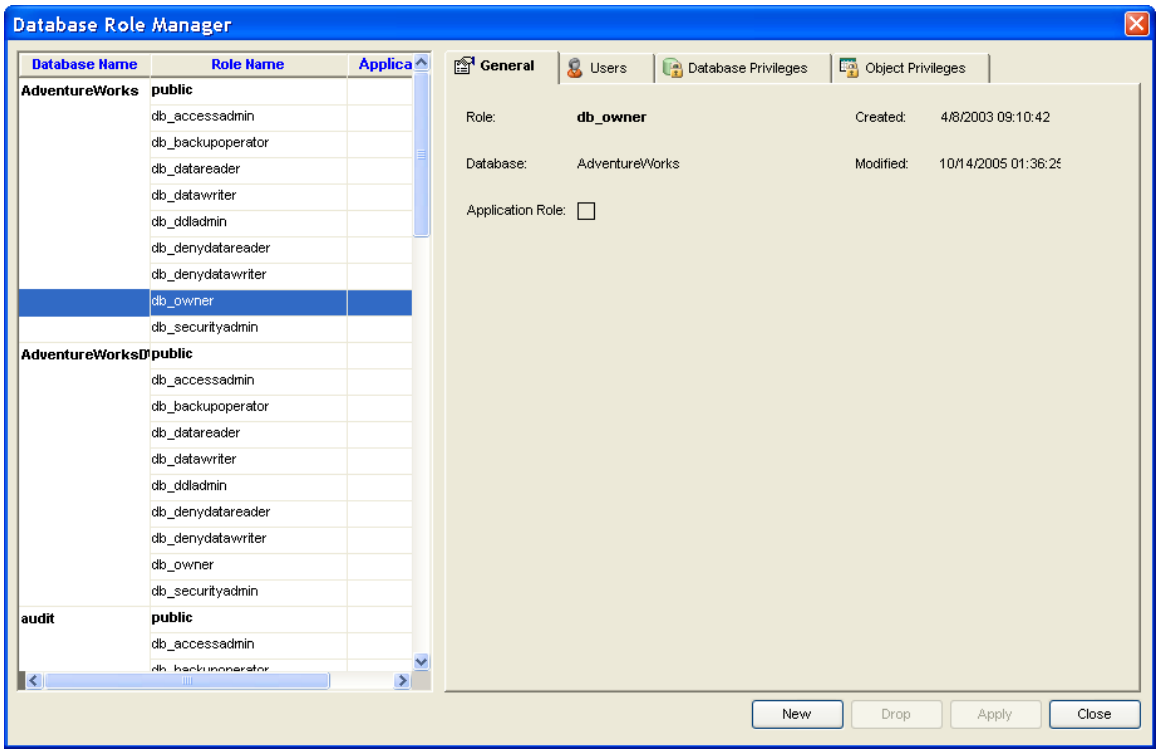
Deleting Database Users

1. Open the **Database User Manager** dialog.
2. On the left hand side of the dialog, select name of the user you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the **YES** button if you are sure that the right user is chosen for deletion.

Managing Database Roles

To start the **Database Role Manager**, click **DB Roles** button on the Login Manager screen.

The **Database Role Manager** dialog consists of two parts: The left hand side is used to list all database roles and their properties in a single table-like list for all databases available on the server. The right hand side is used to display and edit role properties assigned to the selected role.



Database Role Properties

Role – Displays the database role name.

Database – Displays the name of the database in which the role should be defined.

Application Role- Specifies the type of role. A checkmark in the box indicates an application role; otherwise, the role is a standard database role.

Important Note: Application roles work with both authentication modes. Microsoft recommends that you use Windows Authentication when possible.

Created – Displays the date and time the role was created. This property is automatic and cannot be changed.

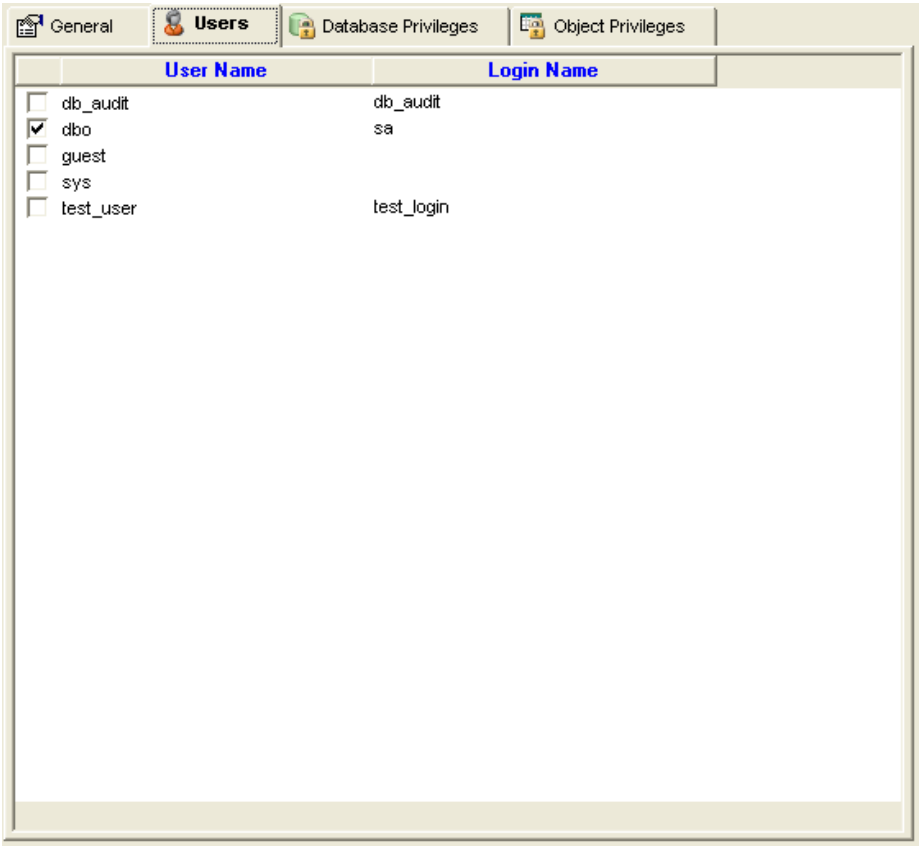
Updated – Displays the date and time the role was last modified last. This property is automatic and cannot be changed.

Granting and Revoking Database and Application Roles

The **Users** tab on the Database Role Manager dialog can be used to control which database users are granted the selected database role. This is just another view to the database user/database role associations. On this screen you can find all database users for a given role while on the Database User Manager screen you can find all roles granted to a given user.

Important Note: When using application roles, you cannot audit individual user activity. You can audit only the activity of the application role.

The behavior and usage of this screen is similar to the behavior of the Database User Manager.



This is a multiple choice screen. You can select as many users as you want. When you click the **Apply** button, all of your changes, including changes on other tabs are applied at once.

To grant the selected role to a user, place a checkmark in the left-most column. To revoke the selected role from a user, clear the corresponding checkmark.

Restrictions: The default **public** role may not be revoked from a database user.

The **total** field shows how many users are currently selected.

Granting and Revoking Database Privileges

The **Database Privileges** tab page on the Database Role Manager dialog can be used to control which database-wide privileges are enabled for the selected role; in other words, which database-wide privileges are granted to the selected role.

The **Database Privileges** tab page behavior and usage is identical to the behavior and usage of the Database Privileges tab on the Database User Manager dialog. Refer to Granting and Revoking Database Privileges topic for instructions on how to use this screen.

This is a multiple choice screen. You can select as many privileges as you want. When you click the **Apply** button, all of your changes, including changes on other tabs are applied at once.

The **total** field shows how many privileges are currently selected.

Important Notes:

User-specific database privileges and role-specific database privileges have equal importance, which may cause various conflicts. For example, if a user **Peter** is granted role **Workers** and that role has DENY privileges for CREATE TABLE operations, but the user is also directly granted CREATE TABLE privilege, **Peter** will be unable to create new tables because of the conflicting permissions.

Server-role privileges supersede database role privileges. For example, consider the case where a login **Domain\Peter** is granted **Server Administrator** role and is also mapped to database user **Peter** who, in turn, is granted the database-specific **Workers** role. In this case, even if the **Workers** role has DENY privileges for CREATE TABLE operations, user **Peter** will still be able to create any table in the database because his login is mapped to top level privileges for the entire server.

Granting and Revoking Object Privileges

The **Object Privileges** tab page on the Database Role Manager dialog can be used to control which database object access privileges are enabled for the selected role, in other words, which privileges are granted to the selected role.

The **Object Privileges** tab page behavior and usage is identical to the behavior and usage of the **Object Privileges** tab on the Database User Manager dialog. Refer to Granting and Revoking Object Privileges topic for instructions on how to use this screen.

This is a multiple choice screen. You can select as many privileges as you want. When you click the **Apply** button, all of your changes, including changes on other tabs are applied at once.

The **total** field shows how many privileges are currently selected.

Important Notes:

User-specific object privileges and role-specific object privileges have equal importance, which may cause conflicts. For example, if a user **Peter** is granted role **Workers** and that role has DENY privileges for **HumanResources.Employee** table, but the user is granted SELECT privileges for **HumanResources.Employee** table, **Peter** user will be unable to access data in that table because of the conflicting permissions.

Server-role privileges supersede database role privileges. For example, if a login **Domain\Peter** is granted **Server Administrator** role and also database-specific **Workers** role and which has DENY privileges for **HumanResources.Employee** table, user **Peter** will still be able to access data in the **HumanResources.Employee** table because of his login mapping with the **Server Administrator** role gives him top level privileges for the entire server.

Creating New Database Roles

1. Open the **Database Role Manager** dialog using **DB Roles** button on the Login Manager screen.

2. Click the **New** button.
3. Fill in new role properties on the **General** tab page.
4. Enter additional options and grant permissions using other tab pages. For detailed information on how to use other tab pages, read previous topics in this chapter.
5. Click the **Apply** button to save changes.

Deleting Database Roles

1. Open the **Database Role Manager** dialog.
2. On the left side of the dialog, select the name of the role you want to delete.
3. Click the **Drop** button. The **Confirm Delete** message will appear. Click the **YES** button if you are sure you have selected the correct role for deletion.

 **Important Note:** All users previously granted the deleted role would lose their database and object privileges associated with the deleted role.

Effective Security Settings

Overview

Using the **Effective Security Settings** utility, you can:

- Analyze effective access rights for any given user, table or an entire schema
- Find out how users gained rights to a particular database table or objects within a particular database schema--whether they accessed them directly or indirectly via assigned roles, group membership, even via database views
- Analyze effective access rights granted to the PUBLIC role.
- Analyze who has access rights to standard database procedures that can be used to access operating system files outside of the database.
- Use the results to ensure enforcement of the security policies

Exploring Effective Security Settings

To launch Effective Security Settings Tools

1. Click **Tools > Effective Security Setting** menu. The following dialog will appear.

2. On the **Goal** page, select the goal for security settings analysis.

   Click the **Next** button.

3. On the **Target** page, select the target database, schema, object or user whose security settings or access you want to check

   Click the **Next** button.

4. On the **Options** page, select the required analysis options.

   Click the Next button.

To save analysis results to an external text file

1. At the top of the **Search Results** page, click the **Save** icon. The **Select Output File** dialog will appear.

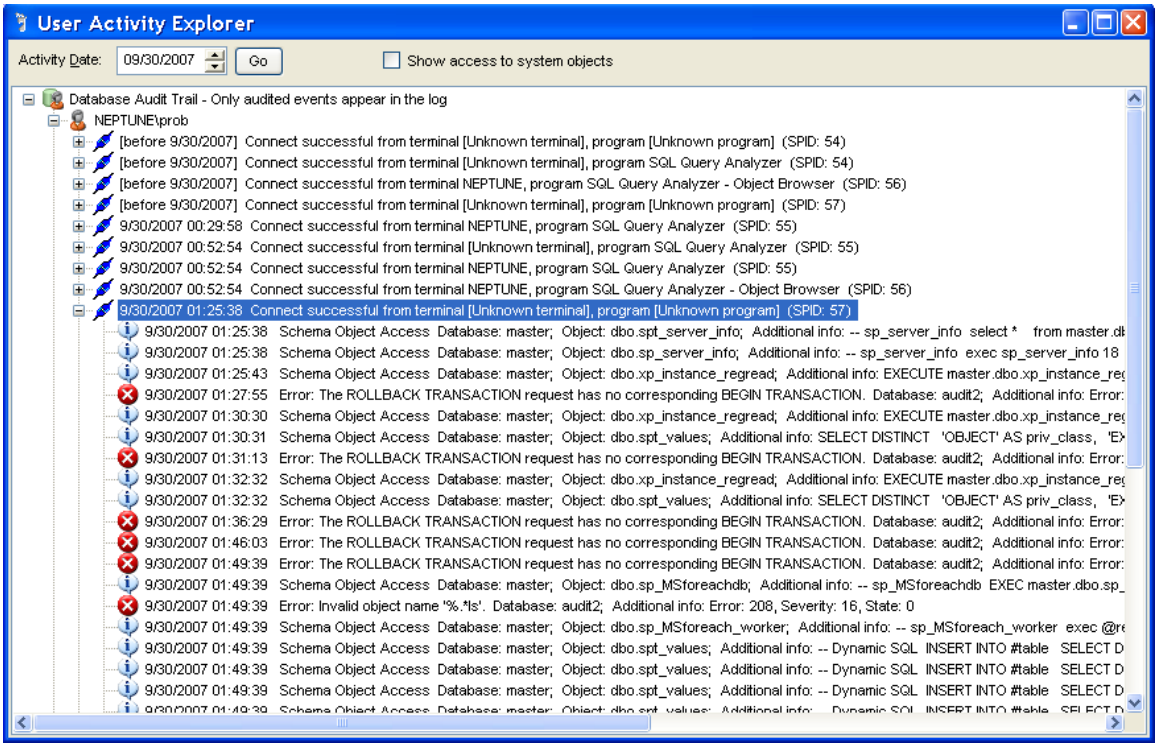
2. Choose the output file name and location and then click the **OK** button on the **Select Output File** dialog.

User Activity Explorer

Overview

User Activity Explorer is a handy tool that can be used for investigative analysis of user activities. Using this tool, you can trace step-by-step activities of any user on any day, provided that auditing has been installed and enabled for audit events generated by the selected user.

Important Note: The level of detail available in the User Activity Explorer depends on the subset of the events chosen for auditing. If you have only chosen to audit logins to the database server, you will not be able to use this tool to figure out what the user did in the database.



Working with User Activity Explorer

1. Click the **Tools > User Activity Explorer** menu or click the **User Activity** button on the DB Audit Management Console toolbar. The **User Activity Explorer** window will appear.

2. Enter the date of the user activities that you want to investigate and press either the Enter key or click the **Go** button. The **User Activity Explorer** window will display all database sessions recorded for the selected date as well as any sessions that started at an earlier time and continued until that date.

3. Locate the database session whose activities you want to drill-down to and either click the [+] sign or double click on the session icon

4. Review the activities. Note that the activities are listed in the order they have been performed. The

contents of the **User Activity Explorer** window depends on the audited events. The display format may slightly vary for different database systems because of the internal differences in the audit event collection and presentation methods.

5. If required, repeat steps 2 to 4 for other users, sessions or dates.

CHAPTER 11: Security Snapshots

Overview

The configuration of your database security and audit settings is subject to change on a regular basis. The Security Snapshots feature of DB Audit can be used to capture point-in-time snapshots of the database server security settings and to store them in the Central Repository database.

The following information can be captured by security snapshots:

- Global database server security and audit settings
- System audit configuration
- Data-change audit configuration
- Database server users and logins
- User/login effective security settings

DB Audit allows you to include in the snapshot only the security and audit configuration data you want to capture.

Snapshots are generated periodically. DB Audit provides a full set of functions for snapshot management, including automatic data collection across multiple database servers, automatic data deletion of old snapshots, and automated error notifications and so on.

- DB Audit also provides several advanced reports for forensic analysis of the collected data as well as reports for side-by-side snapshot comparison. These reports can be used for: tracking and documenting enterprise-wide security changes
- Auditing user access and permissions throughout your organization
- Enforcing security policies
- Self-monitoring.

How it Works

Using the DB Audit graphical or web interface is a three-step process:, First, you register your database servers with the Central Audit Repository system, you then select the security and audit settings categories that you need to capture, and finally you set up a periodical data collection job to capture the changes.

During this set up process, DB Audit creates several tables in the Central Audit Repository database. It also creates a data collection job to periodically populate the repository tables with the security information from registered servers. You can use built-in DB Audit reports to analyze and report on the collected data, or you can create your own custom reports for custom data analysis. All tables storing the collected data are located in the DB_AUDIT schema and all begin with the SNAPSHOT prefix, making them easy to find. Below is a list of tables and descriptions of data stored in each table:

SNAPSHOT_CATALOG – Stores list of servers registered for snapshot data collection and the

individual settings for each server

SNAPSHOT_LOAD_LOG – Stores snapshot processing logs and statistics

SNAPSHOT_DATA_AUDIT – Stores table-level data-change auditing settings

SNAPSHOT_DATA_AUDIT_APPS – Stores application-level filters used by the data-change auditing

SNAPSHOT_DATA_AUDIT_USERS – Stores application-level filters used by the data-change auditing

SNAPSHOT_DATA_AUDIT_COLUMNS – Stores column-level settings used by the data-change auditing

SNAPSHOT_SYS_AUDIT – Stores system-auditing settings

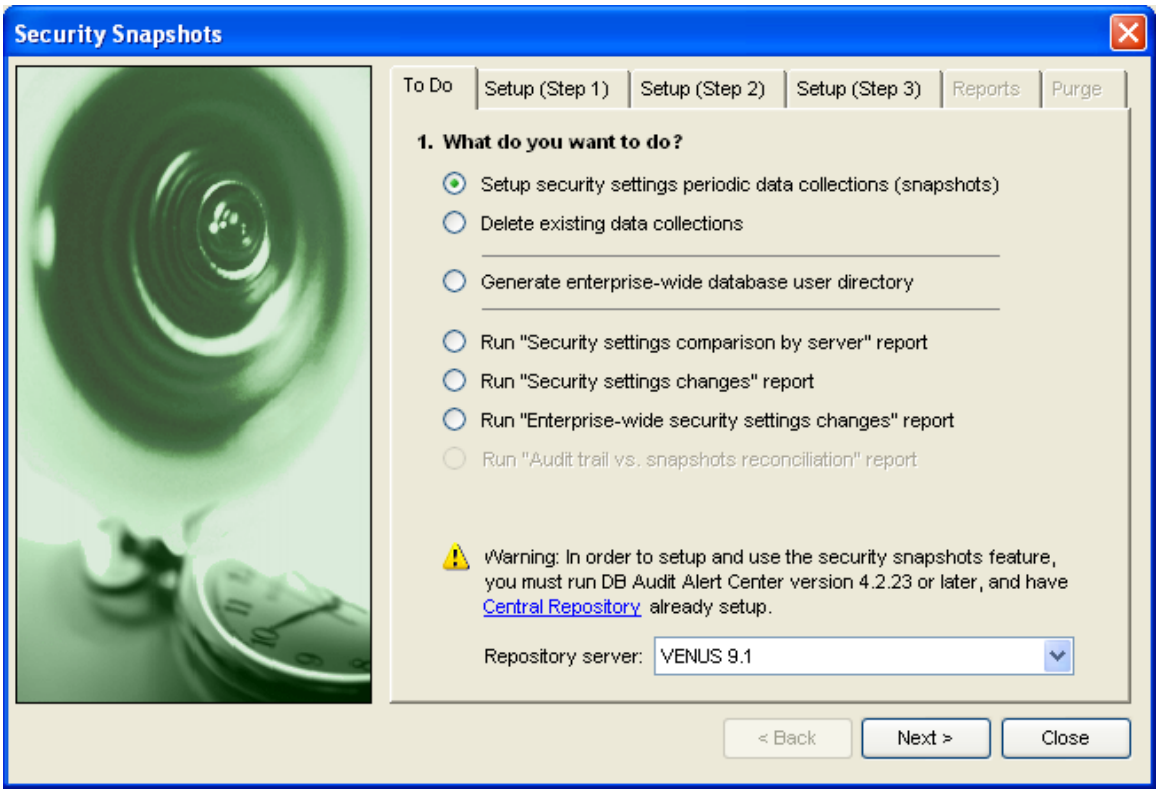
SNAPSHOT_LOGINS – Stores database server users and logins

SNAPSHOT_EFFECTIVE_SECURITY – Stores effective security settings for database server users and logins

SNAPSHOT_SYS_ACCOUNTS – Stores system account names that should not processed by the snapshot data collection job if system accounts are excluded from processing. This table is pre-populated with common system accounts during the installation process. If necessary, you can modify the data in this table directly by deleting unneeded accounts and adding your custom accounts.

Setting up Security Snapshots

Use the **Security Snapshots** tool available in the DB Audit Management Console to register database servers for security snapshots and to configure snapshot settings. This tool can be invoked using **Tools >Security Snapshots** menu.



1. Select the **Setup security settings periodic data collection** option, which is the first available in the **Security Snapshots** wizard.

2. In the drop-down box at the bottom of the **To Do** tab, select the database profile for your repository server. If the repository server hasn't been set up yet, click the Central Repository hyperlink above the drop-down list and set up the Central Repository. See CHAPTER 8, Central Audit Repository for more information on Central Repository setup. Click the **Next** button to continue.

3. On the **Setup (Step 1)** tab, select the types of security and audit configuration data you want to collect. Click the **Next** button to continue.

4. On the **Setup (Step 2)** tab, choose the snapshots data collection job parameters and frequency. Click the **Next** button to continue.

5. On the **Setup (Step 3)** tab, enter email accounts settings that the Alert Center can use to notify personnel in case of Security Snapshot data collection job failures. Click the **Next** button again. The wizard will evaluate the entered setup configuration and will install a new configuration or update an existing security snapshot configuration according to the settings you entered.

Updating Security Snapshots Configuration, Registering and Deregistering Servers

Use the same steps described in the previous topic for updating the existing configuration.

Uninstalling Security Snapshots

Use the following steps to uninstall security snapshots:

1. Start the **Security Snapshots** tool using **Tools >Security Snapshots** menu.
2. Select the **Setup security settings periodic data collection** option, which is the first option available in the **Security Snapshots** wizard.
3. In the drop-down box at the bottom of the **To Do** tab, select the database profile for your repository server. Click the **Next** button to continue.
4. On the **Setup (Step 1)** tab, deselect all types of security and audit configuration data. Click the **Next** button twice. You will get a confirmation message box asking you to confirm that you want to uninstall the security snapshots data collection. Click **Yes** to confirm.

 **Important Notes:** During the Security Snapshots uninstallation process, all previously collected security data will be completely destroyed. That data can be restored only from a database backup.

Manually Deleting Security Snapshots data

This method can be used to manually delete existing security snapshots and free the allocated database space. This method keeps the existing security snapshots configuration intact and future data collections to continue:

1. Start the **Security Snapshots** tool using **Tools > Security Snapshots** menu.
2. Select the **Delete existing data collections** option which is the second option available in the **Security Snapshots** wizard.
3. In the drop-down box at the bottom of the **To Do** tab, select the database profile for your repository server. Click the **Next** button to continue. You will get a prompt to confirm this action. Click **Yes** to confirm.

Generating an Enterprise-wide Database User Directory

To generate a directory, Security Snapshots data collection must be setup and must have been run at least once to run successfully.

Perform the following steps to generate the directory:

1. Start the **Security Snapshots** tool using **Tools > Security Snapshots** menu.

2. Select the **Generate enterprise-wide user directory** option.

3. In the drop-down box at the bottom of the **To Do** tab, select the database profile for your repository server. Click the **Next** button to continue. DB Audit will generate and display the database user directory.

 **Tips:** The Database User Directory is generated as a report that can be printed, sorted, filtered, saved and exported to external files, including HTML, XML, Excel and other formats, just like any other DB Audit report. See Working With Interactive Reports topic in CHAPTER 7 for more information on DB Audit's built-in reporting functions and capabilities.

Auditing and Documenting Security Changes, Enforcing Change Control

The DB Audit **Security Snapshots** tool produces several reports that can be used for auditing and documenting database security changes . The following reports can be used for these purposes.

Security Settings Comparison by Server – This report uses security snapshots to compare side-by-side security settings of two database servers and find the differences. This report can be used for both change control and quality assurance. For example, it can be used to verify that the security and auditing settings on a production server have been configured exactly as they have been configured in a test environment.

Security Settings Changes – This report can be used to investigate and document security and audit settings changes made over time in a particular database server. When running this report, you can choose two specific snapshots that you want to compare.

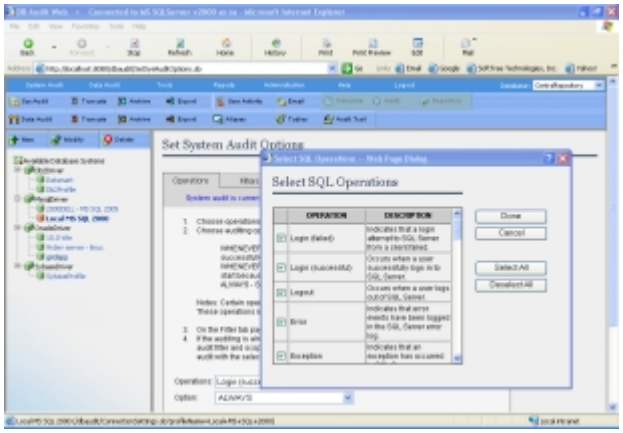
Enterprise-wide security settings changes - – This report can be used to investigate and document security and audit settings changes made over time in all database servers registered for snapshot data collection. When running this report, you can specify the time period for the change analysis. For each server, DB Audit will automatically find the last available snapshot taken before the start of the period and the last available snapshot taken before the end of the period and compare them. Comparison results for all servers will be merged and output on the same report.

CHAPTER 12: Web-based Interface

Overview

The DB Audit Web-based Interface is available with certain types of DB Audit licenses. The web-based interface can be used instead of the graphical DB Audit Management Console and can also be used instead of the Alert Center Remote Console.

The DB Audit Web-based Interface is specifically designed to emulate the look and feel of the DB Audit graphical interfaces for Windows systems. It is menu and toolbar control driven, and it has the same web page dialogs with exactly the same look as the graphical dialogs.



The following features make DB Audit Web-based Interface an attractive option to use in the enterprise:

- As a web-based application, it is easier to deploy and maintain - no need to deal with installations and upgrades on user desktops.
- It provides more flexible and yet tighter security over user access to various auditing functions and reports.
- It runs on, and is accessible from, multiple platforms, including Windows, Linux and Unix
- Supports exporting reports to XML, CSV and TXT formats.
- it allows interface translations to multiple languages.
- As is true of any web application, it can be easily customized and integrated with existing third-party applications.

System Requirements

Web-server:

1. A web server capable of running Java Server Pages, such as Apache Tomcat, IBM WebSphere, MacroMedia JRun; BEA WebLogic and other
2. Java Development Kit (JDK) version 1.4.2 or better
3. At least 50 MB of free disk space on the web server
4. At least 1 GB of RAM or better

Client:

1. A common web browser such as Internet Explorer; Google Chrome, FireFox or compatible web browsers

Configuration Files

The DB Audit Web-based Interface uses four configuration files for storing various settings. The following paragraphs describe these files using the Apache Tomcat server as an example. Similar settings can be used with other supported web servers, taking into account that different web servers use different directory structures for web applications.

1. **web.xml** is a standard, general-purpose web application configuration file that must reside in **[Tomcat directory]/webapps/dbaudit/WEB-INF** directory.
2. **dbaud-users.xml** is a configuration file for storing names and permissions of users working with the DB Audit Web-based Interface. It is highly recommended that you move this file to a secure location accessible to the Apache Tomcat web serve, but not located in a web application directory. This file should not be stored in any location where it can be accessed and modified using any form of web-based interface. For example, the **[Tomcat directory]/conf** directory would be a good place to move this file.
3. **profiles.xml** is a configuration file for storing database connection profiles and their properties. It is highly recommended that you move this file to a secure location accessible by Apache Tomcat web server, but not located in a web application directory where it can be accessed and modified using any form of web-based interface. For example, you can move this file to **[Tomcat directory]/conf** directory.
4. **config.xml** is a configuration file for storing Alert Center connection parameters and license keys. In future versions, the use of this file may be expanded to store other parameters. It is highly recommended that you move this file to a secure location accessible to the Apache Tomcat web server, but not located in a web application directory where it could be accessed and modified using any form of web-based interface. For example, you can move this file to **[Tomcat directory]/conf** directory.

Setting Path to Configuration Files

Before the web application can be used, its **web.xml** file must be updated. The file is located in the **[Tomcat directory]/webapps/dbaudit/WEB-INF** directory. Here is what you want you need to change in that file:

1. Open this file in Windows Notepad, Unix vi or another text editor. Search for the text string**<param-name>dbaudit-users</param-name>**. On the following line, update the value of this parameter by replacing **D:\Work\dbaudit_users** with the actual path to the dbaudit-users.xml file.

2. Search for the string **<param-name>dbaudit-profiles</param-name>**. On the following line, update the value of this parameter by replacing **D:\Work\dbaudit_profiles** with the actual path to the profiles.xml file.

3. Search for the string **<param-name>dbaudit.config.path</param-name>**. On the following line, update the value of this parameter replacing **D:\Work** text with the actual path to config.xml file.

User Access Control

You can assign user access to DB Audit functionality using either the DB Audit graphical interface running on the user's workstation, or using the DB Audit Web-based interface.

DB Audit graphical interfaces support only two types of users: Administrators who have full access to the DB Audit Management Console, and Report Users who have only limited access to audit reports. When installing DB Audit graphical console on a workstation, the type of user access available from that workstation is determined during the installation process. During installation, your choice of components to install on the workstation determines the type of control users can exercise from that workstation.

In contrast, the DB Audit Web-based Interface allows you to configure user access dynamically. The DB Audit Web-based Interface supports three types of user roles which, in turn, determine the type of access users have from within DB Audit Web-based Interface:

- **admin** – Users associated with this role, have access to all functions available in the DB Audit console including access to the **Administration menu,** which can be used to manage other users and their roles.

- **dba** – Users associated with this role have access to nearly all functions available in the DB Audit console, excluding access to the **Administration menu.** DBA users are not granted the ability to manage other web console users. Users having this role can create new database connection profiles and manage existing profiles. They can also set up and reconfigure audit settings for any database for which they have true DBA permissions as assigned on the database server.

- **user**– Users associated with this role have access only to DB Audit reporting functions.

 **Important Notes:** At the initial installation, one user is automatically configured. The name of the user is **admin** and the password is also **admin**. You should logon to DB Audit Management console as that user and immediately change the password and, if desired, the user name. To change the user name and password, use the **Administration** menu available in the console. In the **Administration**

menu, click the **Manage Web Users** submenu to access the user management screens.

CHAPTER 13: Audit Data Retention and Archiving

Data retention policies

Data retention policies are defined by your organization. Do not expect DB Audit to automatically delete old logs, old audit trail data, and so on. Many regulations require that data be retained for at least several years, although the requirements are different for different industries, different countries and even different states with in the same country.

DB Audit provides all the tools you need to set up automatic data purging and audit space recycling, but it is up to your organization to decide on the data retention policies to set up and enforce, where to keep the data, how and when to delete it and whether to keep archived copies of the deleted data or not.

DB Audit offers several methods of data archiving and data purging. All the methods can be used independently or in combination. The following topics describe the available methods and how to use them.

Using the Central Audit Repository for data archiving

The Central Audit Repository can store and consolidate audit data from multiple database systems scattered across the enterprise. It eliminates the need to store vast amounts of the audit trail data on individual audited servers. For detailed information on how to set up and operate the Central Audit repository, see CHAPTER 8, Central Audit Repository.

It is important to remember that the Central Audit Repository and Alert Center setup procedures provide two separate controls for audit data archiving and purging in audited database systems. It also includes an independent control for the periodic data purging in the central repository itself. Do not confuse the controls for purging the central repository with those used to perform automated data archiving and purging on individual audited databases. The methods used on individual audited databases are described in the following two topics.

 **Tips:**

- Alert Center default settings for the Central Audit Repository are set to replicate audit trail data from the audited database to the central repository every 15 minutes and to purge old data from the local audit trails leaving only one month of the data available locally in the audited database. This data is required for certain report types of compliance reporting that need to join results from the audit trails with results stored in the database system catalog tables, such as local user names, their permissions and other local security settings.
- You can modify the default settings according to your business requirements and applicable regulations.
- The central repository purging job is not installed and enabled by default. By default, data is retained forever. If you want to change this setting, use the Central Repository Deployment Tools to set up the purge job. The instructions for setting this job are

available in Central repository audit trail space management topic.

Automatic archiving to files

The DB Audit Management Console can be used to install audit trail archiving procedures for sending audit trail data to operating system files outside of the database. These files can be then compressed using ZIP or other compression utilities and backed up from the database backup.

See the Scheduling periodic audit data trail archiving to files topic for more detailed information on how to setup this process.

Tips:

- Because data in archive files is stored in text formats without indexes and with extra space for future growth, they can be easily compressed into much smaller files, typically providing significant space savings. For example, data in an audit trail table taking 1MB of space within an Oracle database require less than 100 KB of disk storage after archiving and standard ZIP compression.
- Audit data archiving is an incremental process. Only data added to audit trail tables since the previous archiving run is saved to archive files.
- The DB Audit data archiving procedures use date-time suffixes to ensure unique archive file names.
- The main disadvantage of the file based data archiving method is that zipped files are not a convenient source for generating reports. For the archived data to be used in reports, it must be loaded back into the database.

Automatic data purging

The DB Audit Management Console can be used to install audit trail purging procedures for deleting audit trail data that is no longer needed. The freed space can be reused for the new audit data.

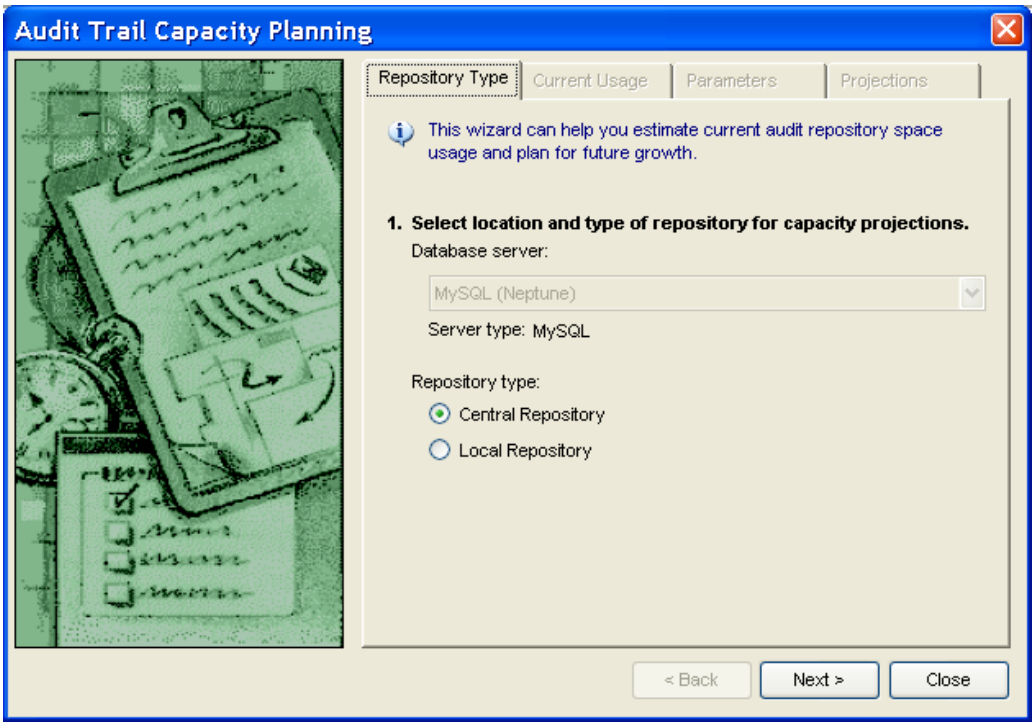
See Scheduling periodic audit trail purge topic for more detail on how to set up this process.

Tip: Automated data purging processes can be installed to run concurrently with automated data archival processes.

Audit trail capacity planning

DB Audit provides an easy to use yet sophisticated utility for analyzing audit trail space usage and for projecting future space requirements. This utility analyzes current data stored in active audit trail tables as well as their indexes and index space. It also analyzes historical usage patterns including the

effects of any data purging and replication processes in place.



This utility can be used to analyze and forecast audit trail requirements for both local and central audit repositories.

Use either of the following methods to begin the analysis process:

- In the DB Audit Management Console main menu, click **Tools > Capacity Planning** menu.
- On the DB Audit Start Page, click the box with **Audit Trail Capacity Planning** label:

The **Audit Trail Capacity Planning** dialog will appear. Note that the database server field is automatically populated with the name of the database profile used for the current database connection.

1. On the **Repository Type** tab, choose the type of repository available on the server. Click the **Next** button to continue.

2. On the **Current Usage** tab, you can review statistics for all currently used audit trail tables. Note that the **Audit Trail Capacity Planning** utility automatically checks space usage statistics for all tables. If there tables with stalled space usage statistics, it automatically runs the required statistics collection processes. If your audit trail tables are large, this statistics collection processing may take a while. Please wait until the processing is complete before clicking the **Next** button. During an active analysis operation, the progress bar appears at the bottom of the DB Audit screen. The progress bar disappears when the analysis process is complete.

3. On the **Parameters** tab, you can enter estimation and data retention parameters for future space requirements, such as the length of the forecasting period, anticipated changes in data auditing, data retention policies, and so on. Enter desired parameter values and then click the **Next** button to generate the space usage projections.

4. Review the report displayed on the **Projections** tab. To print the displayed report, click the **Print** button ( ) displayed above the report view.

CHAPTER 14: Audit Data Adapters and Data Export

Overview

Audit Data adapters provide a way to export collected audit data to external systems or flat files. Processing is run periodically according to a user-defined schedule, and the data is exported incrementally. During each export run, only the most recently added records are exported.

The following types of Data Adapters are available:

- **File Data Adaptor –** This adaptor allows sending audit trail data to operating system files. The operating system files can reside both locally on the Alert Center system and on the remote systems accessible via network shares.
- **SysLog Data Adaptor –** This adaptor allows sending audit trail data to local and remote syslog servers.
- **EventLog Data Adaptor –** This adaptor allows writing audit trail data to local and remote Windows Event Logs. This adaptor can be used when the Alert Center is run on a Windows-based system.
- **SNMP Data Adaptor –** This adaptor allows sending audit trail data to Network Management consoles, such as HP Open View, Microsoft MOM, IBM Tivoli, and many others. The data is sent using SNMP traps.

All Data Adapters can be optionally configured to purge the audit trail data after it has been successfully exported.

Important Notes:

- DB Audit Alert Center is required for running the Data Adapters.
- Data Adapters can be used for exporting results of system auditing. They cannot be used for any other purposes.
- Data Adapters are optional components that are not installed by default. They are provided with the Enterprise License of DB Audit and can be also obtained and licensed separately.

Installing and Configuring Data Adapters

Installing Data Adapters

To install data adapters, unzip **adapters.zip** into the DB Audit Alert Center installation directory.

Configuring Data Adapters – Common Steps

1. **Setting Up New Database Connection**

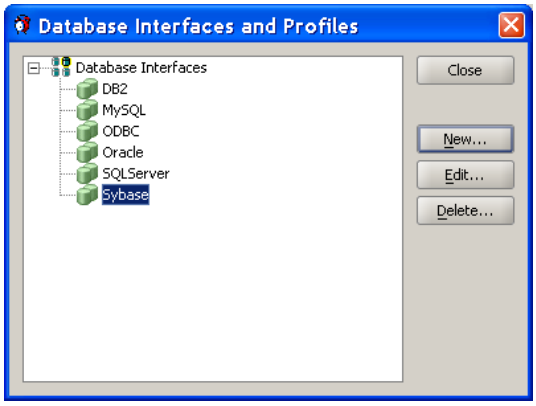
   If you are using Central Audit Repository configuration, make sure the database systems whose audit records you are planning to export have been registered with the Alert Center.

   If you are not using a Central Audit Repository, perform the following manual configuration setup to set up a new database connection in the Alert Center:

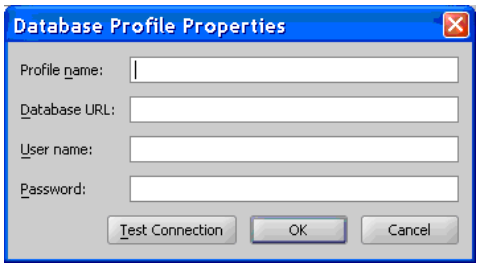
   Start the **Alter Center Scheduler** using master.bat (or ./master.sh) file.

   1.1. On the main screen, click the **Tools > Database Profiles** menu. The **Database Interfaces and Profiles** dialog screen will appear.

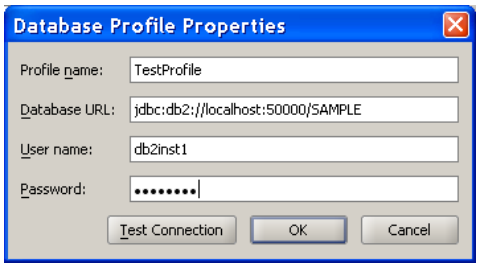
   1.2. Select the database interface compatible with your database system or select ODBC if you would like to use an ODBC connection.



   1.3. Click the **New…** button. The **Database Profile Properties** dialog will appear.



   1.4. Enter database connection information for your database server on which the audit data tables are located. The syntax for Database URKL is specific to the database type. Please consult your database documentation or search the Internet to locate the required syntax. The result may look similar to the following example.



Use the **Test Connection** button to verify that the connection is set up and working properly. Click the **OK** button to close **Database Profile Properties** dialog.

2. **Setting Up New Data Export Job**

2.1. In the Alert Center, click **File > New > Folder** menu. The **Folder Properties Wizard** dialog will appear.

Enter folder name as on the example screenshot. Click the **OK** button to close the dialog.

2.2. In the Alert Center, click **File > New > Job** menu. The **Job Properties Wizard** dialog will appear.

Enter a descriptive job name and click the **Next** button.

2.3. Enter the job command line using one of the following commands:

To use the SysLog Adaptor use:

```
java -jar dbauditAdaptors.jar /JOB @V"job_id" /PROFILE TestProfile
```

/ADAPTOR SysLog

To use the File Adaptor use:
java -jar dbauditAdaptors.jar /JOB @V"job_id" /PROFILE TestProfile /ADAPTOR File

To use the EventLog Adaptor use:
java -jar dbauditAdaptors.jar /JOB @V"job_id" /PROFILE TestProfile /ADAPTOR EventLog

To use the SNMP Adaptor use:
java -jar dbauditAdaptors.jar /JOB @V"job_id" /PROFILE TestProfile /ADAPTOR SNMP

Important Notes: The job command line must be entered as a single line. In the command you must replace the "TestProfile" reference with the name of the database connection profile you configured in step 1. The job command line must be in the following format:
java -jar dbauditAdaptor.jar /JOB <job_id> /PROFILE <profile_name> /ADAPTOR <adaptor_type> [/PURGE] [YES/NO] [/TIMEZONE_OFFSET] [offset]

Note that [/PURGE] [YES/NO] [/TIMEZONE_OFFSET] [offset] are optional parameters whose usage is described later in this chapter.

2.4. Click the **Next** button four times to reach the Job Properties Wizard's step 6 of 11.

For the scheduler's step 6, choose “All day” as the schedule type, then click the **Next** button to advance to step 7.

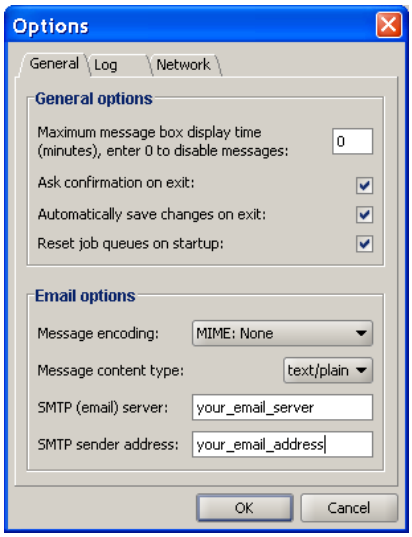
2.5. Change the job frequency as needed. The following screenshot shows the job set to run every 15 minutes.

Click the **Next** button.

2.6. This step is optional and can be used to set up email notifications for failed data export jobs.

For the **Send e-mail message** action/event, select the checkbox in the **Job Error** column, then click the **Next** button.

Enter your email address in the **Account** and in the **Recipients** fields. Click the **system options** hyperlink. The **Options** dialog will appear.



Enter SMTP (email) server and SMTP sender address values, then click the **OK** button to close **Options** dialog.

2.7. On the **Job Properties** Wizard, click the **Next** button again.

Clear the "**Disable this job on error**" checkbox, then click the **Finish** button.

If you need to configure Data Adapters for other database systems, repeat the steps described above starting with the step 1 **Setting Up New Database Connection**

Configuring SysLog Data Adaptor

After completing common steps described in the previous topic you can configure the SysLog adaptor type specific properties.

In the Alert Center directory. Locate the **adapter.properties** file and open it in a text editor. Change the value in **syslog_host** line as required. This is the name or IP address of your local or remote syslog server. Make sure the syslog server is configured to accept remote logging.

Below are three examples of **adapter.properties** files. Note that lines starting with the pound symbol (#) are comment lines.

Example config 1

```
# This example configuration file can be used to generate tab-separated
# syslog records. Each audit record occupies a single line with data values
# separated by tabs
column_headers=false
syslog_facility=AUTH
value_separator=\t
syslog_host=MY_SYSLOG_SERVER
```

Example config 2

```
# This example configuration file can be used to generate
# semicolon-separated syslog records. Each record occupies a single line
# with column headers and data values separated by semicolons
column_headers=true
syslog_facility=AUTH
value_separator=;
syslog_host= MY_SYSLOG_SERVER
```

Example config 3

```
# This example configuration file can be used to generate multi-line
# syslog records with column headers. Each record occupies multiple lines
# with data values appearing on separate lines and each line ending with
# a semicolon
column_headers=true
syslog_facility=AUTH
value_separator=;\n
syslog_host= MY_SYSLOG_SERVER
```

Configuration parameters:

column_headers - this parameter indicates whether or not column headers for audit trail records will be sent along with the audit records data. The parameter value must be one of the following string values: true, false.

syslog_facility – this parameter indicates the SysLog facility name to use when sending the data.

The facility name must be one of the following strings KERN, USER, MAIL, DAEMON, AUTH, SYSLOG, LPR, NEWS, UUCP, CRON, AUTHPRIV, FTP, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7.

AUTH is used by default.

value_separator – this parameter indicates which symbol or symbols will be used to separate values within the audit trail records sent to the SysLog server. Each record is submitted as a single message to the SysLog server and appears in the target log. Each record contains multiple audit data values separated by the values specified in the **value_separator** parameter. The parameter value may contain any printable symbols as well as the following three special values:

\n - end-of-line symbol, ASCII code 10

\r – carriage return symbol, ASCII code 13

\t - tab symbol, ASCII code 9

debug_level – this parameter indicates whether or not the adaptor will write diagnostic messages to the job log file. The parameter value must be one of the following string values: true, false.

log_level – this parameter controls the types of audit records exported from the audit trail. This parameter can be used as a filter. The following types are supported:

INFO – export all records

WARNING – export errors, failed logins, schema changes, security changes, database administration events

ERROR – database errors and failed logins.

If this parameter is not specified, 'INFO' is used by default.

syslog_host – this parameter specified the name or IP address of the syslog server

Configuring EventLog Data Adaptor

After completing the common steps described above, you can configure the EventLog adaptor type specific properties:

In the Alert Center directory, locate the **adapter.properties** file and open it in a text editor. Change the value in the **eventlog_host** line as required. This is the name or IP address of your local or remote Windows Event Log server.

Below are three example **adapter.properties** files. Note that lines starting with # symbol are comment lines.

Example config 1

```
# This example configuration file can be used to generate tab-separated
# event log records. Each audit record occupies a single line with data
# values separated by tabs
column_headers=false
value_separator=\t
eventlog_name=Security
eventlog_host=MY_SERVER
```

Example config 2

```
# This  example configuration file can be used to generate
# semicolon-separated syslog records. Each record occupies a single line
# with column headers and data values separated by semicolons
column_headers=true
value_separator=;
eventlog_name=Security
syslog_host= MY_ SERVER
```

Example config 3

```
# This  example configuration file can be used to generate multi-line
# syslog records with column headers. Each record occupies multiple lines
# with data values appearing on separate lines and each line ending with
# a semicolon
column_headers=true
value_separator=;\n
eventlog_name=Security
syslog_host= MY_SYSLOG_SERVER
```

Configuration parameters:

column_headers - this parameter indicates whether or not column headers for audit trail records will be sent along with the audit records data. The parameter value must be one of the following string values: true, false.

eventlog_name – this parameter indicates in which Windows event log the exported record will appear. If not specified, the default 'Security' log is used.

> The log name must be one of the following strings: :System:, :Application:, :Security:, or the name of a valid custom log file as it appears in the Windows Event Viewer application. If you specify the name of a custom file, the file must be created before DB Audit Data Adapters can use it.

value_separator – this parameter indicates which symbol or symbols will be used to separate values within the audit trail records sent to the Event Log server. Each record is submitted as a single message to the Event Log. Each record contains multiple audit data values separated by the values specified in the **value_separator** parameter. The parameter value may contain any printable symbols as well as the following three special values:

> \n - end-of-line symbol, ASCII code 10
>
> \r – carriage return symbol, ASCII code 13
>
> \t - tab symbol, ASCII code 9

debug_level – this parameter indicates specifies whether or not the adaptor will write diagnostic messages to the job log file. The parameter value must be one of the following string values: true, false.

log_level – this parameter controls the types of audit records exported from the audit trail. This parameter can be used as an export filter. If this parameter is not specified, 'INFO' is used by default.The following types are supported:

> INFO – export all records
>
> WARNING – export errors, failed logins, schema changes, security changes, database administration events
>
> ERROR – database errors and failed logins.

eventlog_host – this parameter indicates the name or IP address of the Event Log server. If not specified, local server is used by default.

Configuring File Data Adaptor

After completing the common steps described above, you can configure the File adaptor type specific properties:

In the Alert Center directory, locate the **adapter.properties** file and open it in a text editor. Change the value in **file_path** line as required. This is the target directory for the export files.

Below are three example **adapter.properties** files. Note that lines starting with # symbol are comment lines.

Example config 1

```
# This example configuration file can be used to generate tab-separated
# text files. Each audit record occupies a single line with data values
# separated by tabs. The output files are automatically compressed.
value_separator=\t
file_path=C:\audit\data\archive
zip_files=true
```

Example config 2

```
# This example configuration file can be used to generate
# semicolon-separated files (CSV). Each record occupies a single line
# with data values separated by commas
value_separator=,
file_path=C:\audit\data\archive
zip_files=false
```

Example config 3

```
# This example configuration file can be used to generate text files
# containing human-readable multi-line audit records with column headers.
# Each record occupies multiple lines with data values appearing on
# separate lines and each line ending with a semicolon
column_headers=true
value_separator=;\n
file_path=C:\audit\data\archive
zip_files=false
```

Configuration parameters:

column_headers - this parameter indicates whether or not column headers for audit trail records will be sent along with the audit records data. The parameter value must be one of the following string values: true, false.

file_path – this parameter indicates to which directory the Adaptor will write output files. The value must be a valid path.

file_name – this parameter can be used to specify custom fixed file names for output files. If not specified, the file name **audit-<time suffix>.txt** is used or, if zip files are used, **zip_data=true**, **audit-<time suffix>.zip** . The **<time suffix>** is replaced with the system time value thus guaranteeing a unique name for each output file.

value_separator – this parameter indicates which symbol or symbols will be used to separate values in the audit trail records. Each record contains multiple audit data values separated by the values specified in the **value_separator** parameter. The parameter value may contain any printable symbols as well as the following three special values:

\n - end-of-line symbol, ASCII code 10

\r – carriage return symbol, ASCII code 13

\t - tab symbol, ASCII code 9

debug_level – this parameter indicates whether or not the adaptor will write diagnostic messages to the job log file. The parameter value must be one of the following string values: true, false.

zip_files – this parameter specifies whether or not the Data Adaptor will compress output files. The parameter value must be one of the following string values: true, false.

Configuring SNMP Data Adaptor

After completing the common steps described above, you can configure the SNMP adaptor type specific properties:

In the Alert Center directory, locate the **adapter.properties** file and open it in a text editor. Change the value in the **snmp_host** line as required. This is the name or IP address of your SNMP Management Console.

Below are three example **adapter.properties** files. Note that lines starting with # symbol are comment lines.

Example config 1

```
# This  example configuration file can be used to generate tab-separated
# syslog records. Each audit record occupies a single line with data values
# separated by tabs
column_headers=false
value_separator=\t
snmp_host=MY_SNMP_CONSOLE_HOST
```

Example config 2

```
# This  example configuration file can be used to generate
# semicolon-separated syslog records. Each record occupies a single line
# with column headers and data values separated by semicolons
column_headers=true
value_separator=;
snmp_host=MY_SNMP_CONSOLE_HOST
```

Example config 3

```
# This  example configuration file can be used to generate multi-line
# syslog records with column headers. Each record  occupies multiple lines
# with data values appearing on separate lines and each line ending with
# a semicolon
```

```
column_headers=true
value_separator=;\n
snmp_host=MY_SNMP_CONSOLE_HOST
```

Configuration parameters:

column_headers - this parameter indicates whether or not column headers for audit trail records will be sent along with the audit records data. The parameter value must be one of the following string values: true, false.

value_separator – this parameter indicates which symbol or symbols will be used to separate values within the audit trail records sent to the SNMP console. Each record is submitted as a single SNMP trap message. Each record contains multiple audit data values separated by the values specified in the **value_separator** parameter. The parameter value may contain any printable symbols as well as the following three special values:

\n - end-of-line symbol, ASCII code 10

\r – carriage return symbol, ASCII code 13

\t - tab symbol, ASCII code 9

debug_level – this parameter indicates whether or not the adaptor will write diagnostic messages to the job log file. The parameter value must be one of the following string values: true, false.

log_level – this parameter controls the types of audit records exported from the audit trail. This parameter can be used as a filter. The following types are supported:

INFO – export all records

WARNING – export errors, failed logins, schema changes, security changes, database administration events

ERROR – database errors and failed logins.

If this parameter is not specified, 'INFO' is used by default.

snmp_host – this parameter specifies the name or IP address of the SNMP server. If this parameter is not specified, 'localhost' is used by default

snmp_port - – this parameter specifies SNMP port used by the SNMP Management Console. If this parameter is not specified, the default SNMP port 162 is used.

 **Tip:**

SNMP traps generated by all SoftTree Technologies products can be uniquely identified by the SoftTree Enterprise OID number **.1.3.6.1.4.1.15277**

SNMP traps generated by DB Audit Data Adapters are identified by the unique OID **.1.3.6.1.4.1.15277.8.1**

Configuring Data Adapters for Data Deletion

DB Audit Data Adapters can be configured to delete processed audit data automatically after the data has been successfully archived and/or transferred to the designated target services.

To enable data deletion, add the **/PURGE YES** command line parameters to the end of the job command line, for example:

```
java -jar dbauditAdaptors.jar /JOB @V"job_id" /PROFILE TestProfile /ADAPTOR SysLog /PURGE YES
```

If this parameter is not specified, data deletion is not performed and the exported data remains on the remote database server.

For more information see the Configuring Data Adapters – Common Steps topic.

Dealing with Time-zone Differences.

Time-zone settings for audited database servers can differ from time-zone settings for the Alert Center system on which DB Audit Data Adapters reside. It is important to configure Data Adaptor jobs to recognize the differences and properly adjust date calculations. Failure to do so, may lead to Data Adapters missing important audit events or waiting for hours before new audit events are noticed.

You can add the **/TIMEZONE_OFFSET <offset>** parameters to the end of the Data Adaptor job command line to indicate the time-zone difference. Here the <offset> placeholder must be replaced with the time difference between the Alert Center system and the remote server from which the audit data will be extracted. This value must be expressed as a number of hours and could be both an integer and a decimal number. Positive numbers are used to specify that time on the remote server is ahead of the time on the Alert Center system. Negative numbers are used to specify that time on the Alert Center system is ahead of the time on the remote server. For example, the following job command line indicates that that Alert Center system is five hours ahead of the database server.

```
java -jar dbauditAdaptors.jar /JOB @V"job_id" /PROFILE TestProfile /ADAPTOR SysLog /TIMEZONE_OFFSET -5
```

If this parameter is not specified, a difference value of zero hours is assumed by default.

For more information see Configuring Data Adapters – Common Steps topic.

CHAPTER 15: Installation and Uninstallation

Front-end Installation

The DB Audit Setup program provides a very straightforward interface for DB Audit installation. Simply follow the Installation Wizard that will guide you through the entire installation process.

If you are going to use an ODBC interface for the database connection, configure the desired ODBC data sources before beginning the installation. Use the ODBC Administrator found in the Windows Control Panel. More details can be found in the Defining the ODBC data source topic.

Back-end Installation

DB Audit Expert will automatically install all required repository objects for the backend. DB_AUDIT user/schema and other optional components are installed when you set up data-change audit triggers or choose certain features that require additional components to be installed. For more information on data-change auditing, refer to the How it works topic.

DB Audit installs the following components:

- DB_AUDIT user – this is required in order to create the DB_AUDIT schema, which contains all the repository objects and service procedures.

  SQL Server, ASE: DB Audit also creates the DB_AUDIT logon, which is required to create the DB_AUDIT user and schema.

  Note: Because the DB_AUDIT user/logon should not be used for normal database connections, the password for this user/logon is intentionally left undocumented. If required by your company's security policy, you can reset the password to any other password of your choice. DB Audit does not make internal connections to the database using the DB_AUDIT user/logon, so the audit processing will not be affected by the password change.

  If you want to change this password before the installation or if the password does not comply with your database system password complexity rules, use the Options screen to change the default password value. See Options topic for more details.

- Repository tables – this includes DB Audit catalog tables for data-change auditing and all audit trails tables. Audit trail tables are created and dropped as needed for the selected data-change auditing configuration. For more information on the data-change auditing, refer to the How it works topic.

- Data-change audit triggers – DB Audit automatically creates and drops audit triggers as required for the selected data-change auditing configuration. For more information on data-change auditing, refer to the How it works topic.

  DB_AUDIT.SP_AUDIT_PURGE – this stored procedure is installed and used if you enable and schedule automatic purge processing for the data-change audit trail.

  Oracle: In addition to purging the data-change audit trail, DB Audit can optionally install the DB_AUDIT.SP_AUDIT_SYSPURGE procedure that can be scheduled to periodically

purge data from the Oracle system audit trail table.

Using the "Advanced Options for Oracle" menu, you can also install a number of other objects for advanced database auditing. For details on how to install and uninstall them, see CHAPTER 3, Configuring Advanced Options for Oracle.

DB2: In some environments, DB2 uses external C compilers for creating SQL stored procedures such as the data audit purge procedure *DB_AUDIT.SP_AUDIT_PURGE* and the system audit purge procedure *DB_AUDIT.SP_AUDIT_SYSPURGE*. To install these procedures successfully, make sure your DB2 server settings are properly configured so that DB2 can locate and use the right compilers. It might be necessary to set the following environment variables:

```
DB2PATH=C:\SQLLIB
DB2_SQLROUTINE_KEEP_FILES=yes
DB2_SQLROUTINE_COMPILER_PATH=your compiler bin directory
```

In addition, you may need to set the correct compiler in the *\SQLLIB\function\routine\sqlproc.mak* file to invoke the correct compiler. For more details, refer to the Scheduling periodic audit-trail purge topic in the Data-change-audit trail management section and also to the Scheduling periodic audit-trail purge topic in the System-audit trail management section.

- **DB2:** If you will be running system-level auditing in DB2, copy the **dbauditRunner.jar** file to your DB2 **[db2 home]/sqllib/function** directory. This file is found in the **/DB2** subfolder of DB Audit Management Console installation folder. Update the CLASSPATH environment variable for the DB2 instance owner and bounce the DB2 server as described in DB2: Enabling system audit topic in CHAPTER 3.

- DP_AUDIT.SP_SEND_MAIL – this stored procedure is installed and used if you set up email support for data-change audit triggers.

  DB2: If you install the DP_AUDIT.SP_SEND_MAIL stored procedure, you must also manually install the **db_audit_mail.class** file. This file is found in the DB Audit installation directory. You must copy this file to the /**sqllib/function** directory of your DB2 instance. The compiled Java class will be used by the DP_AUDIT.SP_SEND_MAIL stored procedure for sending email messages using the Java Mail API.

Alert Center Server Installation

To install and run the Alert Center server, you must have the Java Run-time Environment (JRE) or Java Development Kit (JDK) version 1.4 or better installed on the computer. If you don't have the proper version of JRE or JDK, you can obtain it free of charge from Sun Microsystems http://java.sun.com or from your operating system vendor.

The Alert Center server can be installed and run on the following platforms:

- Windows
- Linux
- Sun Solaris
- HP –UX
- Digital Unix
- IBM AIX
- Free BSD
- Mac OS X

The Alert Center server requires 24x7 Scheduler Alert Center Edition version 2.1 build 121 or later. The 24x7 Scheduler Alert Center Edition can be licensed together with DB Audit as part of the enterprise license, or it can be licensed separately.

Note: The Alert Center and 24x7 Scheduler Alert Center Edition are provided in a single installation package in standard ZIP file format. To install this package, you must first unzip the installation files and copy all unzipped files to the target server.

Installation on Unix/Linux systems

1. If JDK or JRE is not installed on the system or if the installed version is older then 1.4.2, install Java 2 Platform, Standard Edition v 1.4.x or better.
   Note: Make sure your $JAVA_HOME variable points to the right JDK/JRE installation.
2. If Perl is not installed on the system or if the installed version is older then 5.0, install Perl.
3. Copy the installation files to a temporary directory.
4. From that directory, run the installation wizard:
   chmod a+x setup.sh
   ./setup.sh

   Follow the instructions provided by the installation wizard
5. Start the 24x7 Scheduler Alert Center Edition. To start it in graphical mode run ./master.sh. To start it in console mode run ./master.sh nogui.

Tip: If necessary, you can make the 24x7 Scheduler Alert Center Edition start automatically on computer startup. Modify your user startup script and add the command line for the 24x7 Scheduler Alert Center Edition to the script. Use the "nogui" option to have it start in console mode as a background daemon process.

Installation on Windows systems

1. If JDK or JRE is not installed on the system or the installed version is older then 1.4.2, install Java 2 Platform, Standard Edition v 1.4.x or better.
   Note: Make sure your %JAVA_HOME% variable points to the right JDK/JRE installation.
2. Copy the installation files to a temporary directory.
3. From that directory run the installation wizard
   setup.bat

   Follow the instructions provided by the installation wizard.
4. Start 24x7 Scheduler Alert Center Edition. To start it in graphical mode run master.bat. To start it in console mode run master.bat nogui.

Tip: If necessary, you can make the 24x7 Scheduler Alert Center Edition start automatically as a service on computer startup. To start it as a service, first install the 24x7 Scheduler service using the 24x7srv.exe /install command. If necessary, customize the service logon parameters and startup mode using the Services applet in the Windows Control Panel.

Back-end Uninstallation

DB Audit supports standard uninstallation mechanisms for removing program files from the computer.

To uninstall DB Audit database repository objects, data-change audit triggers, data-change audit trail tables, and the DB_AUDIT user/login:

1. Start the DB Audit Expert GUI.
2. Connect to the database server on which you wish to uninstall DB Audit.

   Note: It is highly recommended that you login as a DBA in order to remove the data audit triggers and tables. In SQL Server and ASE, use the SA account. In Oracle, use the SYSTEM or similar account. In ASA use the DBA or similar account, and in DB2, use any appropriate administrator account.

3. Click the **Tools > Uninstall Audit Repository** menu and, when prompted, confirm that you want to uninstall the DB Audit back-end objects, user and settings.

   This will completely remove the DB Audit system from the current server. All tables, triggers and stored procedures installed by DB Audit will be dropped. The DB_AUDIT user and login will be also removed, and all the accumulated audit trail data will be lost. Make a backup first using the appropriate backup utility provided by the DBMS if you want to preserve this data.

Front-end Uninstallation

DB Audit supports standard uninstallation mechanisms for removing program files from the computer.

To uninstall the DB Audit graphical interface:

1. Click the Windows **Start** button. From the **Start** menu, select **Settings**, then **Control Panel**.
2. Double-click **Add/Remove Programs** icon.
3. Select the DB Audit item in the programs list, click the **Add/Remove** button

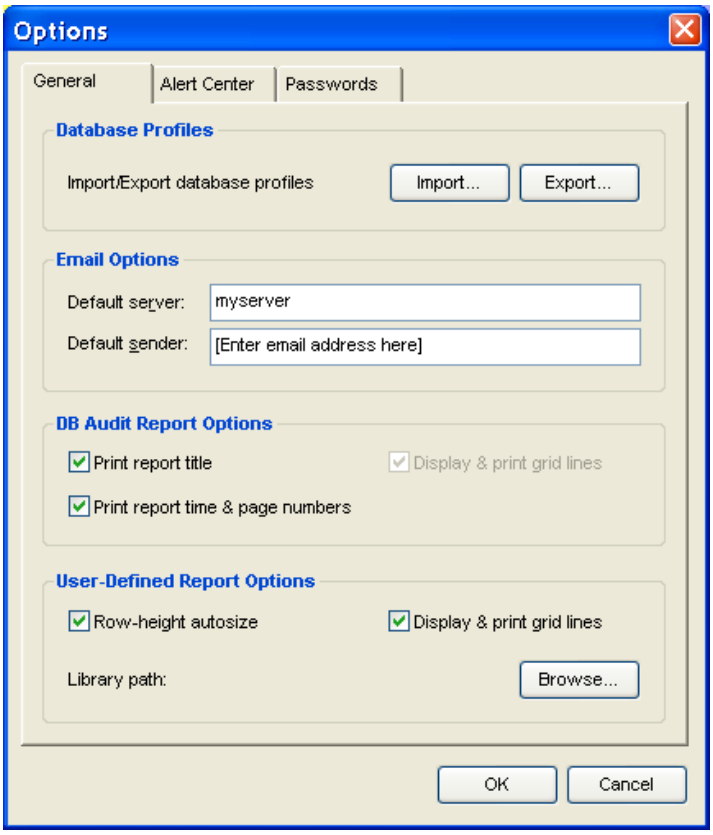
Alert Center Server Uninstallation

Alert Center Server uninstallation is very simple; just delete the directory where you have 24x7 Scheduler Alert Center Edition installed.

Options

Use the **File > Options** menu to access the **Options** screen.

Import/Export functions provide a way to copy audit configuration settings between computers on which DB Audit Expert is installed.



Import – Use this option to import previously exported database profiles.

Export – Use this option to export database profiles configured in the Database Profile Tree to an external file.

Email configuration options provide a way to specify the default email server and recipient address so you don't have to type them in repeatedly when generating triggers for multiple tables with data-change email alerts enabled.

Default server – If the email system on the server is configured to use MAPI, specify the default MAPI profile name in this field. If the email system on the server is configured to use SMTP, specify the email server network name or IP address.

Default sender – Specify the default email address that email alerts will use in to log into the email server.

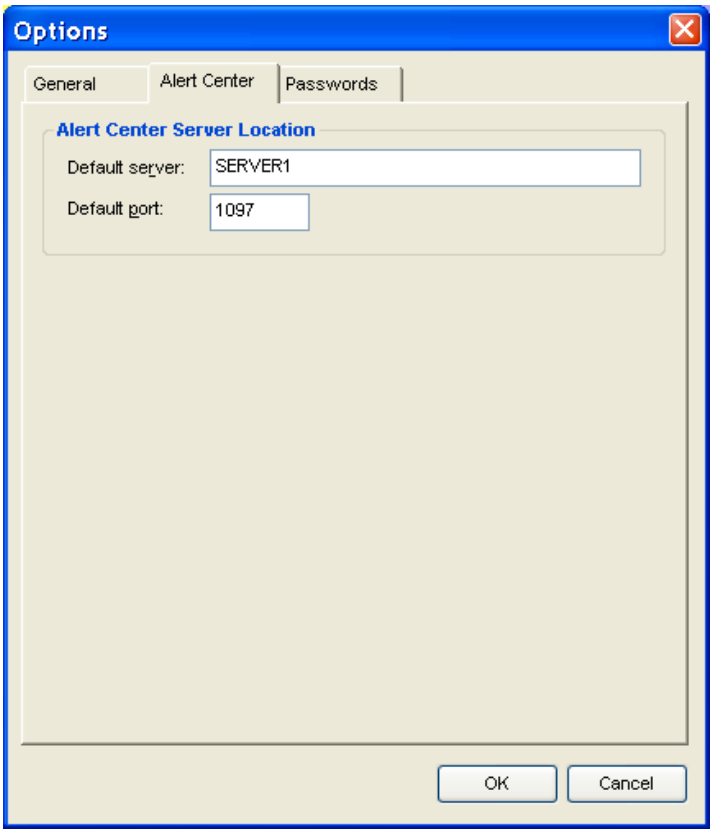
User-defined report options can be used to customize the appearance of generated reports. Note that changing these options takes effect immediately but that the new options do not apply to open reports that are already opened.

Print report title – Select this option if you want the built-in reports to display and print the report title in page headers. Note that when headers are displayed, you cannot rearrange columns on the report using drag-and-drop method.

Print report time and page numbers – Select this option if you want the built-in reports to display and print in page headers the time the report generation time and page numbers. This option can only be used together with **Print report title** option and requires that **Print report title** be also checked.

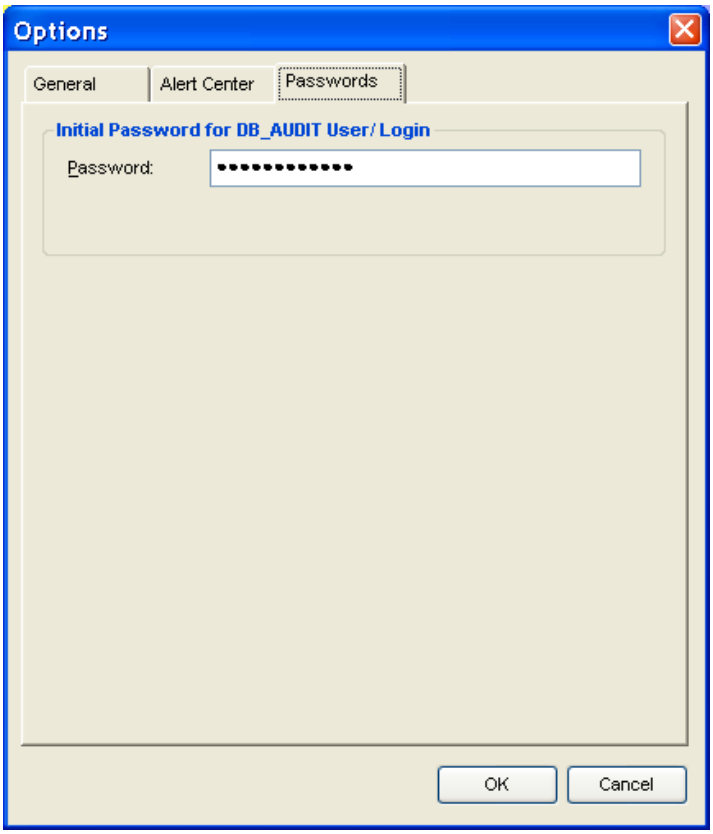
Row-height autosize – Select this option if you want the height of each row in user-defined reports to be determined by the data. If this option is not selected, report rows are assigned a fixed height.

Show grid-lines – Select this option to display grid-lines in user-defined reports. If this option is not selected, grid-lines are not displayed.



Default server – Specifies the name of the default server to which the Alert Center Remote Console and the Central Audit Repository Deployment Tools will attempt to connect.

Default port – Port number on which the Alert Center is configured to listen for network connections.



Password – Specifies the initial password that DB Audit uses when creating DB_AUDIT user and/or login. This password is used for installing system or data-change auditing procedures and repository tables.

CHAPTER 16: Upgrading from Version 3

DB Audit Graphical Management Console

Performing DB Audit upgrades is a very straightforward process. Simply run the DB Audit graphical installation program and follow the prompts displayed by the Installation Wizard. You can choose to install the Management Console into a new folder or into an existing folder for version 3. In either case, all your version 3 settings will be automatically available in version 4.

DB Audit Web-based Management Console

The required upgrade steps depend on the type of the web server you use to run the Management Console. Consult your web-server documentation for information on how to deploy or upgrade a WAR file.

Note that in version 4, DB Audit Web-based Management Console installation has been specifically redesigned to allow seamless upgrades to new versions and maintenance releases. Starting with 4.0, it is no longer required to copy or restore all database profiles and global settings after an update.

The following example describes how to upgrade DB Audit Web-based Management Console running on the Apache Tomcat Web server: The example assumes that the Apache Tomcat Web server is running with the default auto-deploy (also so-called "live-deploy") feature enabled.

1. Locate Tomcat's deployment directory for web-applications **$CATALINA_HOME/webapps/dbaudit/WEB-INF** subdirectory, and find the **config.xml** and **profiles.xml** files in that directory. For example, it might be **C:\Program Files\Apache Software Foundation\Tomcat 5.0.28\webapps\dbaudit\WEB-INF** on Windows or **/usr/local/tomcat/5.0.28/dbaudit/WEB-INF** on Unix or Linux. Copy both XML files to a temporary directory outside of the Tomcat's home directory.

    **Note: dbaudit/WEB-INF** is the default location for **config.xml** and **profiles.xml** files. However, they might reside in some other directory if you changed the default path during installation of the previous version. The path to these files is stored in the **$CATALINA_HOME/webapps/dbaudit/WEB-INF/web.xml** file. You may want to check the location of these files before beginning the upgrade to ensure you are copying the correct files.

2. Copy the **dbaudit.war** file into the **$CATALINA_HOME/webapps/** directory, replacing the existing file.

3. Open your web browser. Connect to the web server and open the DB Audit Web console **http://localhost:8080/dbaudit/.** You will be prompted to generate new configuration files or import the existing configuration files. Choose the "Import config.xml" option and enter the full path to the **config.xml** you copied in step 1. Click the **Ok** button. If the console is unable to find the **profiles.xml** file in the same directory, you might also be asked to specify to the path for that file.

Auditing Service Upgrades

Oracle

If you are running auditing procedures and processes installed with DB Audit 3.1 or 3.2, you do not need to do anything. These procedures are current and forward compatible with DB Audit version 4.

If you use one of the new features available in DB Audit 4 requiring some new back-end procedures to support that feature, DB Audit Management Console automatically installs these procedures after prompting your permissions to proceed with the installation.

Microsoft SQL Server

The system-level auditing procedures have been enhanced in version 4 and a couple of minor bugs have been fixed.

If you are running DB Audit version 3.2, you can simply replace the **xp_dbaudit.dll** installed in the BINN directory of your SQL Server instance with the copy provided in DB Audit version 4. Depending on the type of your SQL Server installation and version, choose the appropriate version of this DLL from the **/MSSQL** subfolder of the DB Audit Management Console installation folder and copy it to the server. Note that in order to replace the existing DLL on the server, you must first stop the SQL Server instance. You can restart the server after the DLL is successfully copied.

If you are running DB Audit version 3.1 or prior, you must temporarily disable system-level auditing on the server and then immediately re-enable it. During this process, the DB Audit Management Console automatically upgrades the existing auditing procedures running on the back-end. For detailed instructions on how to disable auditing and re-enable it, see the Microsoft SQL Server: Disabling system audit and Microsoft SQL Server: Enabling system audit topics in CHAPTER 3. After you disable auditing, make sure to copy **xp_dbaudit.dll** as described above and in CHAPTER3.

Sybase ASE and ASA

If you are running auditing procedures and processes installed with DB Audit 3.1 or 3.2, you do not need to do anything. These procedures are current and forward compatible with DB Audit version 4.

If you use one of the new features available in DB Audit 4 requiring some new back-end procedures to support that feature, DB Audit Management Console automatically installs these procedures after prompting for your permissions to proceed with the installation.

DB2

The system-level auditing procedures have been enhanced in version 4. The new version is backward

compatible with the previous release 3.2. The back-end can be upgraded before or at the same time as the upgrade of the DB Audit Management Console or the DB Audit Report.

If you are running system-level auditing in DB2, copy the new **dbauditRunner.jar** file to your DB2 **[db2 home]/sqllib/function** directory. This file is found in the **/DB2** subfolder of the DB Audit Management Console installation folder.

> **Note:** It is also required to install the **dbauditRunner.jar** file in order to use the PCI, PII and Banking Data Discovery tools with DB2. If you are installing this file first time, don't forget to update the CLASSPATH environment variable for the DB2 instance owner and bounce the DB2 server as described in the DB2: Enabling system audit topic in CHAPTER 3. **You don't have to enable the system auditing for this function to work. Simply follow the instructions provided for updating the environment variables.**

MySQL

Limited experimental support for MySQL auditing has been added in DB Audit version 3.2. If you are running MySQL auditing, follow the steps described in MySQL: Configuring System Audit Options topic in CHAPTER 3 to install the new version.

Alert Center Server Upgrade

The updated version of the Alert Center is shipped with DB Audit 4. If you use the Alert Center, you must upgrade both the DB Audit Management Console and the Alert Center server. Use the following steps to upgrade the Alert Center

1. Shutdown the Alert Center scheduler service.
2. From the Alert Center directory, backup the **DefaultJobDB.dat** and **preferences.xml** files to a temporary directory.
3. Install the new Alert Center version as described in the Alert Center Server Installation topic in CHAPTER 15. You may choose to install it in the same folder where the previous version is installed.
4. Restore the **DefaultJobDB.dat** and **preferences.xml** files in the installation folder.
5. Restart the Alert Center scheduler service.

APPENDIX A: Technical Support

Your questions, comments, and suggestions are welcome.

For technical support, send an email to support@softtreetech.com or use the on-line support form at http://www.softtreetech.com/Support.htm.

Please include your complete contact information when contacting us by email or fax:

Name_____

Company_____

Address_____

City, State/Zip, Country_____

Phone_____Fax_____

Email_____

Best Time to Reach you (specify time zone) _____

Operating System Information:

Which Client operating system are you using: MS Windows 95/98/Me/NT/2000/XP/Vista/Win7 _____

Which DBMS and version are you connecting to: Oracle/SQL Server/ASE/ASA/DB2 _____

What is the host operation system on the computer running your DBMS: _____

When reporting problems, please provide as much information as possible. Be sure to include the following information:

1 Is the problem reproducible? If so, how can this be reproduced? (detail each step)

2 What version of DB Audit Expert are you using?

3 If a dialog box with an error message was displayed, please include the full text of the dialog box, including the text in the title bar.

4 If the problem involves an external program, provide as much information as possible on this program.

Please include the serial number of your copy of DB Audit Expert. Use th Help/About menu to look up the correct numbers. Registered users have priority support.

For registration information, purchasing, or other sales information, please contact our sales department: sales@softtreetech.com or dial 800 289-9256.

For general information, software updates, the latest information on known problems, and answers to frequently asked questions visit the DB Audit home page on the Web: http://www.softtreetech.com/dbaudit/.

We're happy to help in any way we can, but if you are having problems, please check the DB Audit FAQ section first to see if your question is answered there.

APPENDIX B: Hardware and Software Requirements

Minimum Requirements

Client (DB Audit Management Console)

1 Intel or AMD-based workstation or server running one of the following operating system:
- Windows 2012
- Windows 2008
- Windows 7
- Windows Vista
- Windows 2003
- Windows XP
- Windows 2000
- Windows NT 4.0
- Windows 95
- Windows 98
- Windows Me

2 At least 512 MB RAM

3 40 MB disk space for full installation

4 VGA monitor

5 Required database client software (consult your database system documentation for details)

6 If ODBC database interface is used, ODBC and ODBC database connectivity driver

Alert Center (server)

1 Workstation or server running one of the following operating system:
- Windows 2012/2008/7/Vista/2003/Windows XP/Windows 2000/Windows NT 4.0
- Linux – Debian and compatible distributions, RedHat Linux and compatible distributions, SuSe and other
- Sun Solaris
- HP –UX
- Digital Unix
- IBM AIX
- Free BSD
- Mac OS X
- z/OS
- OS/390

2 At least 256 MB RAM

3 18 MB disk space for full installation

4 VGA or other monitor

5 JRE or JDK 1.4

6 Required database client software (consult your database system documentation for details)

7 If ODBC database interface is used, ODBC and ODBC database connectivity driver

Database Server

Any of the supported database servers:

- Oracle 7.3, 8.0, 8i, 9i, 10g, 11g
- Microsoft SQL Server 7, 2000, 2005, 2008, 2012
- Sybase SQL Server and Sybase Adaptive Server Enterprise 10.x, 11.x, 12.x, 15.x
- Sybase Adaptive Server Anywhere 6, 7, 8, 9, 10, 11
- IBM DB2 7.x, 8.x, 9.x , 10.xfor Linux, Unix, and Windows
- IBM DB2 6.x, 7.x, 8.x, 9.x for z/OS and OS/390
- IBM DB2 5.x for OS/400, MVS
- MySQL 4.2, 5.x for Linux and Windows

APPENDIX C: Licensing

LICENSE TYPES

Management Console Single Server License: (one Workstation & one Server)

Permits the license holder to install and use DB Audit Expert Management Console on a single workstation and to use that installation to manage and view auditing on a single local or network-connected database server.

Report Viewer Single User License: (one Workstation & Unlimited Servers)

Permits the license holder to install and use DB Audit Expert Report Viewer application on **one** workstation, and to use that installation to run built-in reports as well as to create, modify and run user-defined audit reports on an unlimited number of network-connected database servers.

Site License: (Unlimited Workstations & Unlimited Servers)

Permits the license holder to install and use DB Audit Expert on an unlimited number of workstations within one geographical location such as an office or building, and to manage and audit an unlimited number of network-connected database servers.

Enterprise License: (Unlimited Workstations & Unlimited Servers)

Permits the license holder to install and use DB Audit Expert on an unlimited number of workstations within one organization in any number of geographical locations, and to manage and audit an unlimited number of network-connected database servers.

SOFTWARE PRODUCT SINGLE USER LICENSE

Copyright laws and international copyright treaties, as well as other intellectual property laws and treaties, protect this SOFTWARE PRODUCT. The SOFTWARE PRODUCT is licensed, not sold.

CAUTION: Loading this software onto a computer indicates your acceptance of the following terms. Please read them carefully.

GRANT OF LICENSE: SoftTree Technologies, Inc. ("SoftTree Technologies") grants you a license to use the software ("Software"). One licensed copy of the Software may either be used by a single person who uses the software personally on one or more computers, or installed on a single workstation used non-simultaneously by multiple people, but not both. One licensed copy of the Software can be used with any number of database servers.

You may make other copies of the Software for backup and archival purposes only. You may permanently transfer all of your rights under this Software LICENSE only in conjunction with a permanent transfer of your validly licensed copy of the product(s).

LICENSE TYPES: The Software and associated add-in components are licensed on a RUN-TIME basis, which means that for each computer on which the Software is installed, a valid run-time license must exist.

Database Management Console License
Allows installation and execution of the Software on a single computer (a stand-alone computer or a single workstation in a network or a single network server) per license for the purpose of installing and managing database audit objects and settings on a single database server as well as running ad-hoc

audit reports.

Report Viewer License
Allows installation and execution of the Software on a single computer (a stand-alone computer or a single workstation in a network or a single network server) per license for the purpose of running ad-hoc audit reports.

Site License
Allows installation and execution of the Software on multiple computers within a single physical location (i.e. an office or data center location at a single physical address).

Enterprise License
Allows installation and execution of the Software on multiple computers in multiple locations throughout the licensed company's facilities.

RESTRICTIONS: Unregistered versions (shareware licensed copies) of the Software may be used for a period of not more than 30 days. After 30 days, you must either stop using the Software or purchase a validly licensed copy.

You must maintain all copyright notices on all copies of the Software. You may not sell copies of the Software to third parties without the express written consent of SoftTree Technologies and under SoftTree Technologies' instruction.

EVALUATION copies may be distributed freely without charge so long as the Software remains whole, including but not limited to, inclusion of existing copyright notices, installation and setup utilities, help files, and the licensing agreement, In executing an act such as distributing the software without the copyright notices is a license violation. To the maximum extent permitted by law, you may be held liable for loss of revenue to SoftTree Technologies or SoftTree Technologies' representatives due to loss of sales or devaluation of the Software or both.

You must comply with all applicable laws regarding the use of the Software.

COPYRIGHT: The Software is the proprietary product of SoftTree Technologies and is protected by copyright law. You acquire only the right to use the Software and do not acquire any rights of ownership.

For your convenience, SoftTree Technologies provides certain Software components in source code format. You may customize this code for your environment, but you must agree not to publish, transfer, or redistribute in any other form either the original code or the modified code.
You must also agree not to remove any product identification, copyright notices, or other notices or proprietary restrictions from the Software.

You must agree not to cause or permit the reverse engineering, disassembly, or decompilation of the Software. You shall not disclose the results of any benchmark tests of the Software to any third party without SoftTree Technologies' prior written approval.

DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS: You may not rent, lease or transfer the Software except as outlined under GRANT OF LICENSE - use and copy.

Without prejudice to any other rights, SoftTree Technologies may terminate this Software LICENSE if you fail to comply with the terms and conditions of this Software LICENSE. In such event, you must destroy all copies of the Software and all of its component parts.

WARRANTY DISCLAIMER: SoftTree Technologies provides this license on an "as is" basis without warranty of any kind; SoftTree Technologies disclaims all express and implied warranties, including the implied warranties of merchantability or fitness for a particular purpose.

LIMITATION OF LIABILITY: SoftTree Technologies shall not be liable for any damages, including direct, indirect, incidental, special or consequential damages, or damages for loss of profits, revenue, data or data use, incurred by you or any third party, whether in an action in contract or tort, even if you or any other person has been advised of the possibility of such damages.

SoftTree Technologies, Inc.
Ilyce Ct 62,
Staten Island NY, 10306
USA