**CERT-In**

**Indian Computer Emergency Response Team**

*Enhancing Cyber Security in India*

# Guidelines for Auditing and Logging

**Department of Information Technology**
**Ministry of Communications and Information Technology**
**Government of India**

**Version 2** **Date: 31st December 2008**

## Table of Contents

# 1. Introduction

Systems and network device reporting is important to the overall health and security of systems. Every network device, Operating System or Applications provide logging features. Log provides a clear view of who owns the process, what action was initiated, when it was initiated, where the action occurred and why the process ran? Thus, it is utmost important that most information should be logged in log files.

**Log is a record of actions and events that takes place on a computer system. Logs are the primary record keepers of system and network activity. When security controls experience failures, logs would be particularly helpful.**

It is in the best interest of organizations to have appropriate auditing policies in place that effectively and efficiently collect the information regarding events including critical events occurring in the network and systems, in the form of logs and manage them appropriately. This information regarding events, available in the form of logs, would enable CIOs, Network and System Administrators to assess:

- Establishing baselines
- Performance of various servers & systems
- Application’s functional and operational problems
- Effectively detect intrusion attempts
- Forensic analysis
- Comply with various regulatory laws

## 1.1 Purpose and Scope

This guideline attempts to provide some insights into the issues related to Auditing and Log Management and suggest best practices for enabling and maintaining Auditing and logging on Windows hosts, Linux hosts, Microsoft IIS Web server, Apache Web server, Oracle 10g Database Server and Microsoft SQL Server 2005. Implementation of these best practices will enable administrators to acquire vital information to identify and respond to the computer security incidents.

## 1.2 Log Analysis for Incident Management

1.2.1 The logs provide vital inputs for managing the computer security incidents, both for Incident Prevention and Incident Response.

1.2.2 When responding to a computer security incident, while the symptoms provide preliminary inputs, logs lead to the activities performed over the system.

1.2.3 During the Incident Response process, logs would be analysed for finding out, the servers and applications that were targeted, such as web-server, mail server, database server etc, the type of attack launched, services and ports that were used, IP addresses of the targeted hosts, source of attack origin (IP address), time stamps etc.

1.2.4 After analyzing and correlating the logs obtained from various devices in the network such as firewall, IDS/IPS, Application servers (web, mail and database), systems etc, requisite details could be obtained and conclusions be drawn about the type and time of incident, vulnerabilities exploited etc.

1.2.5 Similarly, the Incident Prevention process could also gain vital inputs from logs and appropriate countermeasures could be taken to mitigate the risk. Timely analysis of logs would provide important information in the form of precursor to the attacks, such as network scanning attempts, probing queries, information gathering activities (reconnaissance) etc.

## 1.3 Logs for Forensics and Investigation

1.3.1 The logs collected from network devices, servers and systems need to be preserved for facilitating appropriate investigation, assist the Law Enforcement and to take appropriate legal action against the attackers.

1.3.2 The logs would provide vital clues about the attacks & attackers and would enhance the quality of evidence.

# 2. Important considerations for Auditing and Log Management

Auditing is the formal examination and review of actions taken by system users. Event auditing allows the reliable, fine-grained, and configurable logging of a variety of security-relevant system events, including logins, configuration changes and file & network access. These log records can be invaluable for live system monitoring, intrusion detection, and postmortem analysis.

The audit mechanisms and log management can generate, maintain, and protect an audit trails. The following rationales are to be considered while implementing auditing and logging on the web server, application server or any network device:

## 2.1 Audit Policy and Log Generation

Audit policy detects and deters penetration of any organisation's computer system and reveals the usage that identifies misuse. Audits may be conducted to:

- Ensure confidentiality, integrity, and availability by reviewing and maintaining audit logs.
- Investigate possible security incidents to reconstruct the sequence of events that preceded a problem and everything that occurred after it. This reconstruction can assist investigators to assess the full extent of an intrusion which may be required to bring forward any legal action. In this event, the way the log data is handled may make or break the case.
- Ensuring regulatory compliance or compliance to organisation's security policy
- Monitor user or system activity where appropriate, to prevent unauthorized accessing or disclosure of any sensitive information.
- Ensure the handling of sensitive information is restricted to authorized personnel and systems logs are established and maintained on its access.
- Ensure that access to audit logs is restricted and a separation of duties is maintained.
- Ensure that Audit logs are reviewed on a recurring basis with the ability to do selective auditing of information by exception.
- Ensure audit logs are archived for future references.

It is important that appropriate Audit policies should be formulated and implemented on all computer systems, servers and network devices to facilitate collection of logs about all important events. A lapse in this area might result in improper post-incident analysis besides hampering associated Forensic Analysis and investigation.

After the initial setup and configuration of the logging process, system administrator needs to verify that logging is active and working perfectly. Most importantly, all systems and logs should have synchronized clocks; otherwise, timestamps will be inappropriate. This can be achieved by synchronizing internal clocks of all systems with a hardened timeserver on the network.

## 2.2 Challenges in Log Management

In a typical organisation, various hosts with different Operating Systems, Security systems such as Firewall, IDS etc, and other applications generate and store voluminous logs. Different log sources types use unique formats for their logs such as OS-specific formats (e.g., syslog[1] on UNIX systems, event logs on Windows systems) and storage types like text files; databases; and proprietary file formats. This complicates log management because logs are generated from multiple systems spanning across the organisation.

To facilitate log analysis, organisations often need to implement automated methods of converting logs with different formats to a single standard format. Inconsistent log formats also present challenges to people reviewing logs, who need to understand the meaning of each data field in each log to perform a thorough review.

This challenge can be addressed by using products like syslog or Security Information and Event Management (SIEM). (Refer "Common Syslog Server Implementations" [68-75] and "Common SIEM products" [76-89] in Section 11: References and Resources)

## 2.3 Log Storage and Protection

Best Practices for Log Storage and Protection are:

- All log collection and consolidation efforts should be performed on an independent and dedicated log server.
- Contents of the log data should be properly encrypted for protection and digitally signed to ensure integrity.
- Log files need to be set to "append only" to avoid deletions, purges, and overwrites. A good practice would be to have the logs written to a WORM (Write Once Read Many) device, such as a CD/DVD; this way, accidental deletions are prevented via physical means.
- Regular backups of all log files should be conducted at scheduled intervals (daily, weekly, monthly, and so forth) and follow a naming convention conveying information about the date, type, server, and anything else that may be relevant. Integrating the log backup with the overall corporate backup strategy would be beneficial.
- Log files can easily become tremendous in size if set to monitor every detail. Sometimes, this is considered a burden; however, with storage capacity costs decreasing at an incredible rate, and compliance issues requiring immense audit trails, it might be easier to log all events, and have tools available to filter out and bring key events to our attention.
- Secure disposal policies defined by the organization should be used when wiping and shredding log data and media.

[1] Syslog is typically used for computer system management and security auditing. It is a standard for forwarding log messages in an IP network.

- Regular management reports should be generated to properly track backup and disposal events and detect any anomalies that might arise. This organization of data will allow an investigator to track intrusions and discover when incidents appeared.

## 3. System Logs

System logs contain the local events logged by the system or host such as Microsoft Windows XP or Linux system. It helps in identifying normal and abnormal entries. It’s an important resource for computer system management. Collection of important logs on host system would enable the administrators to:

- Look for performance issues of Operating System and Applications
- Detect the malicious and unauthorized activities such as failed Login attempts, data theft, malicious activities performed by virus, worms, bots etc.

Collection and securing the important logs from the host is vital in the security process. Methodologies for enabling auditing, generation of logs and managing the same are discussed in the following sections.

In order to provide individual user accountability, the computing system must be able to correctly identify and authenticate each user. This is the distinguishing factor between system log data and user audit data. Log data, captured by *syslogd* for example, is typically generated by system processes and daemons that report significant events or information. It neither corresponds to specific user actions nor directly traceable to a specific user. However, audit data generated by the system corresponds directly to recorded actions taken by identifiable and authenticated users. All the audit information gathered must be sufficient for an after-the-fact investigation. In a sense, audit data is the complete recorded history of a system user.

Administrators that review the audit data must watch for events that may signify misuse or abuse of the system and user privileges or intrusions. Some examples include:

- Accessing files requiring higher privilege
- Killing system processes
- Opening a different user's files, mail, etc.
- Probing the system
- Installing of unauthorized, potentially damaging software (backdoors, Trojan Horses, etc.)
- Exploiting a security vulnerability to gain higher or different privileges
- Modifying or deleting sensitive information

# 4. Auditing and Logging for Windows Hosts

## 4.1 Audit Policy

Audit policy pertains to specific categories of security-related events that an administrator wants to audit. By default, all auditing policy settings are not defined. ***On domain controllers, auditing is turned off by default***. By turning ON various auditing event categories, administrator can implement audit policy that suits the security needs of the organization.

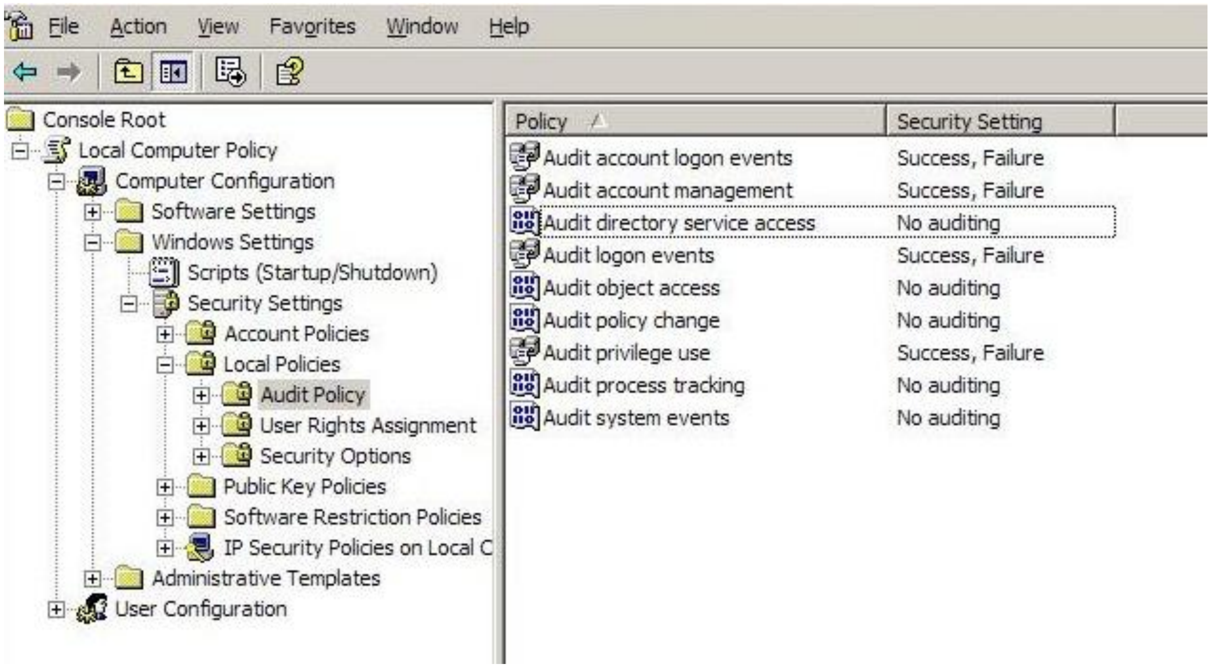
An audit log records an entry whenever user performs certain specified actions. For example, the modification of a file or a policy can trigger an audit entry that shows the action that was performed, the associated user account, and the date and time of the action. Both successful and failed attempts action can be audited.

Security audits are extremely important for any enterprise network, as audit logs may provide an indication that a security breach has occurred. If the breach is discovered some other way, proper audit settings will generate an audit log that contains important information about the breach. If no audit settings are configured, it will be difficult or impossible to determine what occurred during a security incident. However, if audit settings are configured so that too many authorized activities generate events, the *Security Event Log* will be filled with useless data.

Regular security analyses enable administrators to track and determine that adequate security measures are in effect for each computer as part of an enterprise risk management program. Such analyses focus on highly specific information about all aspects of a computer related to security, which administrators can use to adjust the security levels. More importantly, this information can help detect any security flaws that may occur in the computer over time.

Before any audit processes are implemented, an organization should determine how they will collect, organize, and analyze the data. There is little value in large volumes of audit data if there is no underlying plan to utilize it. Also, audit settings can affect computer performance. The effect of a given combination of settings may be negligible on an end-user computer but quite noticeable on a busy server. Therefore, it is recommended to perform some performance tests before deploying new audit settings in the production environment.

Audit policy settings can be configured at following location within the Group Policy Object Editor:
Computer Configuration\Windows Settings\Security Settings\Local Policies \Audit Policy



| Policy | Security Setting |
|---|---|
| Audit account logon events | Success, Failure |
| Audit account management | Success, Failure |
| Audit directory service access | No auditing |
| Audit logon events | Success, Failure |
| Audit object access | No auditing |
| Audit policy change | No auditing |
| Audit privilege use | Success, Failure |
| Audit process tracking | No auditing |
| Audit system events | No auditing |

Figure-1: Group Policy Editor Settings for Audit Policy

The options available for each of the audit settings are:

| Security Setting | Description |
|---|---|
| Success | An audit entry is generated when the requested action succeeds |
| Failure | An audit entry is generated when the requested action fails |
| No Auditing | No audit entry is generated for the associated action |

Table-1: Audit Settings options

### 4.1.1 Audit account logon events

This security setting determines whether to audit each instance of a user logging on to or logging off from another computer in which this computer is used to validate the account. Account logon events are generated when a domain user account is authenticated on a domain controller. The event is logged in the domain controller's security log. Logon events are generated when a local user is authenticated on a local computer. The event is logged in the local security log. Account logoff events are not generated.

If an administrator configures this policy setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when an account logon attempt succeeds, which is useful information for accounting purposes and for post-incident forensics so that who successfully logged into which computer can be determined. Failure audits generate an audit entry when an account logon attempt fails, which is useful for intrusion detection.

However, this policy setting also creates the potential for a Denial-of-Service (DoS) attack. When the "*Audit: Shut down system immediately if unable to log security audits"* setting is enabled, an attacker could generate millions of logon failures, fill the Security event log, and force the computer to shut down.

If an administrator configures the Audit account logon events setting to Success on a domain controller, an entry is logged for each user who is validated against that domain controller, even though the user actually logs on to a workstation or server that is joined to the domain.

### 4.1.2 Audit account management

This policy setting determines whether to audit each account management event on a computer. Examples of account management events include the following:

- A user account or group is created, changed, or deleted
- A user account is renamed, disabled, or enabled
- A password is set or changed

If an administrator configures the Audit account management setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when any account management event succeeds, and user should enable them on all computers in the enterprise. When an organization responds to security incidents, it is critical that they should be able to track who created, changed, or deleted an account. Failure audits generate an audit entry when any account management event fails. To set the value to No auditing, click Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.

### 4.1.3 Audit directory service access

This policy setting determines whether to audit user access of an "Active Directory" directory service object that has an associated System Access Control List (SACL[2]).

If an administrator configures the Audit directory service access setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when a user successfully accesses an Active Directory object that has a SACL that indicates that the user should be audited for the requested action. Failure audits generate an audit entry when a user unsuccessfully attempts to access an Active Directory object that has a SACL that requires auditing. (Both types of audit entries are created before the user is notified that the request succeeded or failed.) To set value to No auditing, click Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.

[2] A SACL is list of users and groups for which actions on an object are to be audited on a Microsoft Windows–based network.

If an administrator enables this policy setting and configure SACLs on directory objects, a large volume of entries can be generated in the Security logs on domain controllers. An Administrator should only enable these settings if actually intend to use the information that is created.

Note: Administrator can configure a SACL on an Active Directory object through the Security tab in that object's Properties dialog box. This method is analogous to Audit object access, except that it applies only to Active Directory objects and not to file system and registry objects.

### 4.1.4 Audit logon events

This policy setting determines whether to audit each instance of user logon, logoff, or network connection to the computer that records the audit event. Account logon events are generated on domain controllers for domain account activity and on local computers for local account activity. If system administrator logs successful account logon audit events on a domain controller, workstation logon attempts do not generate logon audits. Only interactive and network logon attempts to the domain controller itself generate logon events on the domain controller. To summarize, account logon events (described in section 4.1.1) are generated where the account lives, and logon events are generated where the logon attempt occurs.

If an administrator configures the Audit logon events setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when a logon attempt succeeds, which is useful information for accounting purposes and for post-incident forensics so that administrator can determine who successfully logged on to which computer. Failure audits generate an audit entry when a logon attempt fails, which is useful for intrusion detection. To set value to No auditing, click Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.

This configuration also creates a potential DoS condition, because an attacker could generate millions of logon failures, fill the Security event log, and force the server to shut down. Hence suitable log rotation methodology should support the auditing.

### 4.1.5 Audit object access

This policy setting determines whether to audit the event of a user who accesses an object—for example, a file, folder, registry key, or printer—that has a SACL that specifies a requirement for auditing.

If an administrator configures the Audit object access setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when a user successfully accesses an object that has a SACL. Failure audits generate an audit entry when a user unsuccessfully attempts to access an object that has a SACL (some failure events are to be expected during normal computer operations). For example, many applications (such as Microsoft Word) always attempt to open files with both read and write privileges. If the applications are unable to do so, they then try to open the files with read-only privileges. If user enables failure auditing and the appropriate SACL on the file, a failure event will be recorded when such an event occurs.

To set the value to No auditing, click Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.

In Microsoft Windows Server 2003 with Service Pack 1 (SP1), an administrator can audit access to objects that are stored in the Internet Information server (IIS) metabase. To enable metabase object auditing, administrator must enable Audit object access on the target computer, and then set SACLs on the specific metabase objects whose access he wants to audit.
If an administrator configures the Audit object access policy setting and configure SACLs on objects, a large volume of entries can be generated in the Security logs on computers in the organization. Therefore, administrator should only enable these settings if actually intend to use the information that is logged.

Note: Administrator must perform a two-step process to enable the capability to audit an object, such as a file, folder, printer, or registry key, in Windows Server 2003. After administrator enables the audit object access policy, he must determine the objects whose access he wants to monitor, and then modify their SACLs accordingly. For example, if administrator wants to audit any attempts by users to open a particular file, he can configure a Success or Failure audit attribute directly on the file that he wants to monitor for that particular event with Windows Explorer or Group Policy.

## 4.2 Audit policy change

This policy setting determines whether to audit every incidence of a change to user rights assignment policies, Windows Firewall policies, Audit policies, or trust policies.

If an administrator configures the Audit policy change setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when a change to user rights assignment policies, Audit policies, or trust policies is successful. This audit information is useful for accounting purposes and can help to determine who successfully modified policies in the domain or on individual computers. Failure audits generate an audit entry when a change to user rights assignment policies, Audit policies, or trust policies fails. To set the value to No auditing, click Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.

If an administrator enables the Audit policy change setting in Windows XP with SP2 and Windows Server 2003 with SP1, logging of configuration changes for the Windows Firewall component is also enabled.

## 4.3 Audit privilege use

This policy setting determines whether to audit each instance of a user who exercises a user right.

If an administrator configures the Audit privilege use setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when the exercise of a user right succeeds. Failure audits generate an audit entry when the exercise of a user right fails. To set this value to No auditing, in the Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.

If an administrator enables this policy setting, the volume of events that is generated can be very large and the events may be difficult to sort through. This setting should only be enabled when there is a plan to use the information that is generated.

Audit events are NOT generated for use of the following user rights, even if success audits or failure audits are specified for this policy setting:

- Bypass traverse checking
- Debug programs
- Create a token object
- Replace process level token
- Generate security audits
- Backup files and directories
- Restore files and directories

## 4.4 Audit process tracking

This policy setting determines whether to audit detailed tracking information for events such as program activation, process exit, handle duplication, and indirect object access.

If an administrator configures the Audit process tracking setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when the process being tracked succeeds. Failure audits generate an audit entry when the process being tracked fails. To set the value to No auditing, click Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.

If an administrator enables Audit process tracking in Windows XP with SP2 and Windows Server 2003 with SP1, Windows will also log information about the operating mode and status of the Windows Firewall component.

When enabled, the Audit process tracking setting generates a large number of events. This policy setting is typically configured to No Auditing. However, the information that this policy setting generates can be very beneficial during an incident response because it provides a detailed log of the processes that were started and when they were launched.

## 4.5 Audit system events

This policy setting determines whether to audit when a user restarts or shuts down their computer, or when an event occurs that affects either computer security or the Security log.

If an administrator configures the Audit system events setting, he can specify whether to audit successes, audit failures, or do not audit the event type at all. Success audits generate an audit entry when an event executes successfully. Failure audits generate an audit entry when an event is unsuccessful. To set the value to No auditing, click Properties dialog box for this policy setting, select the Define these policy settings check box and clear the Success and Failure check boxes.
As few additional events are recorded if both failure and success audits are enabled for system events, and because all such events are very significant, administrator should configure this policy setting to Enabled on all computers in the organization.

## 4.6 Event Logs

The Event Log records events on the computer and the Security log records audit events. There are three basic logs in Windows environment, these are:

- **Application Logs**: The application log contains events logged by applications or programs. For example, a database program might record a file error in the application log.
- **Security Logs**: The security log records events such as valid and invalid logon attempts, as well as events related to resource use such as creating, opening, or deleting files or other objects. For example, if logon auditing is enabled, attempts to log on to the system are recorded in the security log.
- **System Logs**: The system log contains events logged by Windows system components. For example, the failure of a driver or other system component to load during startup is recorded in the system log. The event types logged by system components are predetermined by the server.

Application and system logging start automatically when the computer starts. Security log starts logging as per the audit policy defined either locally or domain-wide (in Active Directory implementations).

Apart from the logs mentioned above, Microsoft Windows also make other “custom log records” according to the applications or services installed on the system such as Internet Explorer logs, Windows Powershell logs, DNS logs, Directory Service logs, File replication Service logs etc. Again, the connotation of all these types of logs are to record all the events occurring on the system which can be application specific or system specific.

## 4.7 Event Viewer

Using Event Viewer, user can view and set logging options for event logs in order to gather information about hardware, software, and system problems.

The event header contains the following information:

| **Information Type** | **Description** |
|---|---|
| Date | The date the event occurred. The date and time of an event are stored in Universal Time Coordinate (UTC) but always display in the viewer's locale. |
| Time | The time the event occurred. The date and time of an event are stored in UTC but always display in the viewer's locale. |
| User | The name of the user on whose behalf the event occurred. This name is the client ID if the event was actually caused by a server process or the primary ID if impersonation is not taking place. Where applicable, a security log entry contains both the primary and impersonation IDs. Impersonation occurs when the server allows one process to take on the security attributes of another. |
| Computer | The name of the computer where the event occurred. This is usually the name of user’s own computer, unless user is viewing an event log on another computer. |
| Source | The software that logged the event, which can be either a program name such as SQL Server, or a component of the system (such as a driver name) or of a large program. For example, "Elnkii" indicates an EtherLink II driver. The Source always remains in its original language. |
| Event | A number identifying the particular event type for this source. The first line of the description usually contains the name of the event type. For example, 6005 is the ID of the event that occurs when the Event log service is started. The first line of the description of such an event is "The Event log service was started." Using the values of Source and Event together, product support representatives can troubleshoot system problems. |
| Type | A classification of the event severity: Error, Information, or Warning in the system and application logs; Success Audit or Failure Audit in the security log. In the Event Viewer normal list view, these are represented by a symbol. |
| Category | A classification of the event by the event source. This information is primarily used in the security log. For example, for security audits, this corresponds to one of the event types for which success or failure auditing can be enabled in Group Policy by a member of the Administrators group. |

Table-2: Event Header Information Types

## Different types of events:

| Event type | Description |
|---|---|
| Error | A significant problem, such as loss of data or loss of functionality. For example, if a service fails to load during startup, an Error will be logged. |
| Warning | An event that is not necessarily significant, but might indicate a possible future problem. For example, when disk space is low, a Warning might be logged. |
| Information | An event that describes the successful operation of an application, driver, or service. For example, when a network driver loads successfully, an Information event will be logged. |
| Success Audit | Any audited security event that succeeds. For example, a user's successful attempt to log on the system will be logged as a Success Audit event. |
| Failure Audit | Any audited security event that fails. For example, if a user tries to access a network drive and fails, the attempt will be logged as a Failure Audit event. |

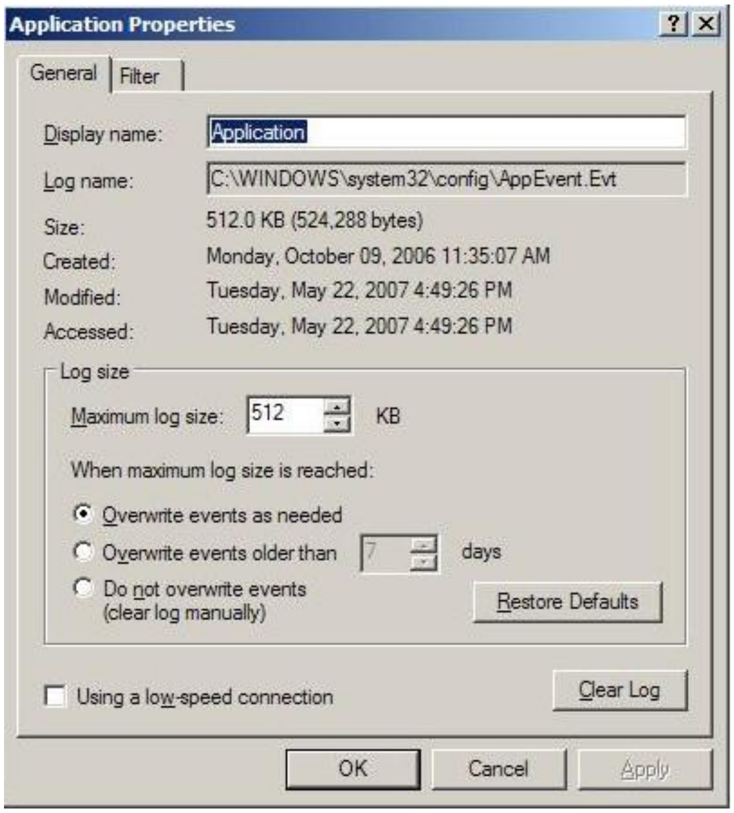
Table-3: Event Types

## 4.8 Configuring logging parameters

Administrator can use Group Policy to set the maximum log sizes and log wrapping options, as well as set access permissions on event logs. Configure the event log settings in the following location within the Group Policy Object Editor:

Computer Configuration\Windows Settings\Security Settings\Event Log\Settings for Event Logs
These event log settings do not appear in the Local Computer Policy object.

Another way to define logging parameters for each kind of event log is by using Event Viewer. To define parameters, click start, type “eventvwr” in the RUN window, “Event Viewer” Management console will be opened, right-click a log in the console tree, and then click Properties. On the General tab, administrator can set the maximum size of the log and specify whether the events are overwritten or stored for a certain period of time (see Figure-2)



Application Properties

General | Filter

Display name: Application

Log name: C:\WINDOWS\system32\config\AppEvent.Evt

Size: 512.0 KB (524,288 bytes)

Created: Monday, October 09, 2006 11:35:07 AM

Modified: Tuesday, May 22, 2007 4:49:26 PM

Accessed: Tuesday, May 22, 2007 4:49:26 PM

Log size

Maximum log size: 512 KB

When maximum log size is reached:

[✓ Overwrite events as needed]

Overwrite events older than 7 days

Do not overwrite events (clear log manually)

Using a low-speed connection

Figure-2: Application Log Properties

### 4.8.1 Settings for “Maximum log size”

This security setting specifies the maximum size of the event log of any type application, security or system which has a maximum limit of 4 GB.

Log file sizes must be a multiple of 64 KB. If an administrator enters a value that is not a multiple of 64 KB, Event Viewer will set the log file size to a multiple of 64 KB.

Even though the theoretical limit for the Event Viewer and Group Policy Object Editor UIs allows to specify as much as 4GB per log—Microsoft has verified that the practical limit is approximately 300 MB on most servers-- that is, 300 MB for all of the event logs combined. On Microsoft Windows XP, member servers, and stand-alone servers, the combined size of the Application, Security, and System event logs should not exceed 300 MB. On domain controllers, the combined size of these three logs plus the Active Directory® directory service, DNS, and Replication logs should not exceed 300 MB.

These limitations can be prevailed over by using central-logging servers like *syslog* or by using SIEM (Security Information and Event Management) products.

Event Log size and log wrapping should be defined to match the business and security requirements, determined at the time of designing the enterprise security plan.

Although there is no simple equation to determine the best log size for a particular server, administrator can calculate a reasonable size. The average event takes up about 500 bytes within each log, and the log file sizes must be a multiple of 64 KB. If administrator can estimate the average number of events that are generated each day for each type of log in organization, user can determine a good size for each type of log file.

For example, if a file server generates 5,000 events per day in its Security log and user wants to ensure that he should have at least 4 weeks of data available at all times, then user would want to configure the size of that log to about 70 MB. (500 bytes * 5000 events/day * 28 days = 70,000,000 bytes.) Then, check the servers occasionally over the following four weeks to verify calculations and that the logs retain enough events for user’s needs.

### 4.8.2 Retaining event logs

This policy setting determines the number of days of event log data to retain for the Application, Security, and System logs; if the retention method that is specified for the log is By Days. Configure this setting only if administrator archive the log at scheduled intervals and make sure that the maximum log size is large enough to accommodate the interval.
The possible values for the “Retain event logs” settings are:

- A user-defined number in days between 1 and 365
- Not Defined (in Group Policy Object Editor)

A user must be assigned the “Manage auditing and security log” user right to access the Security log.

### 4.8.3 Retention method for event log

The default logging policy is that if a log is full, the oldest events are deleted to make room for new events, provided events are at least seven days old. Administrator can customize this policy, or use event log wrapping options, for different logs.

Event log wrapping options include the following:

| **Use** | **To** |
|---|---|
| Overwrite events as needed | Have new events continue to be written when the log is full. Each new event replaces the oldest event in the log. This option is a good choice for low-maintenance systems. |
| Overwrite events older than [x] days | Retain the log for the number of days administrator specifies before overwriting events. The default is seven days. This option is the best choice if administrator wants to archive log files weekly. This strategy minimizes the chance of losing important log entries and at the same time keeps log sizes reasonable. |

| **Use** | **To** |
|---|---|
| Do not overwrite events | Clear or archive the log manually rather than automatically. Select this option only if administrator cannot afford to miss an event (for example, for the security log at a site where security is extremely important). |

Table-4: Event Log Options

In addition to the above options available for the event log settings in Group Policy Object Editor, one more option is available for “Retention method for event log”, that is “Not Defined” (for the purpose when administrator do not want to define this parameter in the Group Policy Object Editor).

# 5. Auditing and Logging for Linux Hosts

## 5.1 Linux System Logging

By default Linux system logs are contained in the directory **/var/log**. Some applications such as ***httpd*** and ***samba*** have a directory within /var/log for their log files. Most log files are in plain text format. Some log files are readable by all users on the system; however, root privileges are required to read most log files. The log files being in text format is readable by any text editor.

## 5.2 Syslog daemon

The syslog program is a distributed logging interface providing a standardized framework under which programs (both operating system and applications) can issue messages to be stored either on the local system or sent to a remote host.

Logging on Linux system can be done through the ***syslogd***, syslog daemon.

Syslogd listens for messages from either port 514 (UDP) or through /dev/log. Once running, almost any part of the system, including applications, drivers, as well as other daemons can make log entries.

Syslog accepts log data from the kernel, from all local processes and even from processes on remote machines.

## 5.3 Configuring *syslogd*, syslog daemon

Syslog configuration consists of routing error messages from various facilities (and at various severity levels) to one or more of the following destinations:

- Logfiles anywhere on the system
- Another computer running syslog with its own syslog configuration
- Active users on the system

Default configuration file for syslogd is located at **/etc/syslog.conf**
The configuration file contains one rule per line. Each rule consists of selector portion, which determines the events to react to and the action portion, which determines what is to be done.
Thus Syslog’s mapping of actions to facilities and priorities are specified in **/etc/syslog.conf.**

The selector has the general syntax:
facility.priority/level
The facility part says what aspect of the system is to be recorded and the priority says what level of messages to react to.

### 5.3.1 Facility

The facility argument is used to specify what type of program is logging the message. The developer of a program decides which facility a program will utilize. In some cases, it may be configurable by the end user. Messages from different facilities can be handled differently by editing the syslog configuration file.

### Supported facilities in Linux Platform:

| | |
|---|---|
| **LOG_AUTH** | security/authorization messages (DEPRECATED Use **LOG_AUTHPRIV** instead) |
| **LOG_AUTHPRIV** | security/authorization messages (private) |
| **LOG_CRON** | clock daemon (**cron** and **at**) |
| **LOG_DAEMON** | system daemons without separate facility value |
| **LOG_FTP** | ftp daemon |
| **LOG_KERN** | kernel messages |
| **LOG_LOCAL0** through **LOG_LOCAL7** | reserved for local use |
| **LOG_LPR** | line printer subsystem |
| **LOG_MAIL** | mail subsystem |
| **LOG_NEWS** | USENET news subsystem |
| **LOG_SYSLOG** | messages generated internally by **syslogd** |
| **LOG_USER** (default) | generic user-level messages |
| **LOG_UUCP** | UUCP subsystem |

Table-5: Supported Facilities in Linux

### 5.3.2 Priority/Level

The log-level or priority determines the importance of the message. The levels are in order of decreasing importance.

| | |
|---|---|
| **LOG_EMERG** | system is unusable |
| **LOG_ALERT** | action must be taken immediately |
| **LOG_CRIT** | critical conditions |
| **LOG_ERR** | error conditions |
| **LOG_WARNING** | warning conditions |
| **LOG_NOTICE** | normal, but significant, condition |
| **LOG_INFO** | informational message |
| **LOG_DEBUG** | debug-level message |

Table-6: Priority Levels

### 5.3.3 An example /etc/syslog.conf

An example syslog configuration file /etc/syslog.conf with description of each setting therein is given below.

```
kern.*                                                  /dev/console
*.info;mail.none;news.none;authpriv.none;cron.none      /var/log/messages
authpriv.*                                              /var/log/secure
mail.*                                                  /var/log/maillog
cron.*                                                  /var/log/cron
*.emerg                                                 @loghost
*.emerg                                                 *
uucp,news.crit                                          /var/log/spooler
local7.*                                                /var/log/boot.log
news.=crit                                              /var/log/news/news.crit
*.emerg;kernel.critical                                 root
```

**Description:**

```
kern.*                                                  /dev/console
```

Kernel messages are sent to the console.

```
*.info;mail.none;news.none;authpriv.none;cron.none      /var/log/messages
```

All messages of priority "info" and above are logged in the /var/log/messages file, but none from the mail, news, cron or authentication facilities/subsystems.

```
authpriv.*                                              /var/log/secure
```

Messages from authentication facility are logged to file /var/log/secure

```
*.emerg                                                 @loghost
```

Sends all emergency messages to the remote machine “loghost”

*Note: The IP address of @loghost is defined in /etc/hosts file so as to enable the system to send the logs to the remote machine. Also the remote host’s syslogd process will need to be enabled for remote logging described in section*

```
uucp,news.crit                                  /var/log/spooler
```

Messages of priority “info” and above for uucp and news are logged to file /var/log/spooler.

```
news.=crit                                    /var/log/news/news.crit
```

Messages of priority “crit” only for news are logged to file /var/log/news/news.crit file

```
*.emerg;kernel.critical                       root
```

This is a multiple selector on a single line. In this case all emergency messages as well as critical messages from the kernel facility would be notified to the user root.

```
*.emerg                                       *
```

Messages of priority emergency and above are notified to all logged in users.

## 5.4 Running syslogd

By deafault syslog daemon is run at the system start up.

On most of the linux systems the start up script for syslogd is located at **/etc/init.d/syslog.**

## 5.5 Remote logging

Almost all distributions of Linux servers have a syslog daemon by default. However the default configuration of the syslog daemon of Linux doesn’t accept syslog messages from the network. In order to tell syslogd to accept remote logs, remote logging must be turned on in two configuration files. The two configuration file settings are similar, so the same change will be performed on both files. Modify the files **/etc/init.d/syslog** and **/etc/sysconfig/syslog**.

Add –r in the SYSLOGD_OPTIONS so that the lines are as shown below.

SYSLOGD_OPTIONS = “ –m 0 –r –x ”

Save and exit the file and restart syslog using the command
***service syslog restart***

## 5.6 Syslog-ng ( new generation )

Syslog daemon *syslogd* logs over User Datagram Protocol (UDP) port 514. UDP is connectionless and does not guarantee reliable communication. Moreover its setup can’t tunnel the messages through stunnel or ssh, thus making it less secure.

Syslog-ng (new generation) is a centralized syslogging solution developed by BalaBit IT Security. It provides a centralized, securely stored log of all devices on the network, independent of the platform.

Some of the advantages of syslog-ng over syslog are:

- The ability to transport syslog messages over TCP along with UDP.
- Filtering based on message contents
- Support for encryption
- Ability to run in a chroot-ed environment.

Detailed guidelines for Implementation of syslog-ng can be referred from **CERT-In Security Guideline CISG-2004-03 "Implementing Central Logging Server using syslog-ng"**

# 6. Web Server Logs

Web logs store a lot of useful information, and being able to sift through that information and find key events is a crucial part of log analysis. Since log files are the most critical source of data for most of information technology systems, they need to be properly managed. There are several areas to focus on when managing log files including conversion, rotation, archival, and separation.

If patterns of normal activity are kept, unusual activity will stand out. The absence of an adequate logging system for Web applications may result in undetected unauthorized accesses that might blind an organization to security threats and breaches.

Depending on the amount of traffic to the Web site, the size of the log file (or the number of log files) can consume valuable disk space, memory resources, and CPU cycles. Administrator might need to balance the gathering of detailed data with the need to limit files to a manageable size and number. If an administrator is planning to put many web sites on one Web server with high traffic volumes and disk writes, he might want to use centralized binary logging to preserve server resources. Also, consider limiting log size by changing the frequency of log file creation. Facility is to be created to record events for each application and Web site on the Web server. Separate logs for each of the applications and Web sites can be created.

Typically, a web server log records all the activities it performs. The W3C maintains a standard format for web server log files, but other proprietary formats also exist. Although formats can differ, the typical page request entry in a web server log contains the following:

- Information about the URL that was requested
- user's IP address
- time and date that the page request was completed
- The referrer header
- HTTP code, bytes served and user agent

All these parameters are important from log analysis perspective and can also assist in cyber security investigations.

Apart from security perspective, a web server logs facilitates:

- To determine which sites on web server or portions of web sites are most heavily viewed, and configure web server to optimize performance accordingly.
- To detect trends such as growth patterns in user visits and plan how to upgrade the hardware and software on web server machine to accommodate such growth in the future.

It is vital to set Web servers and their host systems to keep proper logging. If using Apache, it is recommended to set the log settings to NCSA Combined format; if using Internet Information Server (IIS), set the log settings to W3C Extended format. This will ensure the Web server logs contain the most information. Additionally, the host system should be set properly to log system and security events. Be sure to set the host to log maximum events to catch easily all the events possible.

The auditing and logging procedures for Microsoft IIS web server and Apache web server are covered in Sections 7 and 8 respectively.

# 7. Auditing and Logging for Microsoft IIS Web Server

In earlier versions of IIS, web logs were managed by the "IIS Admin Service" (a service). In IIS 6, web logging using W3C Extended, IIS, and NCSA formats is handled by the HTTP Listener (Http.sys) for better performance and to avoid concurrency issues. Web logging using ODBC, however, is managed by the worker process (w3wp.exe) associated with the site.

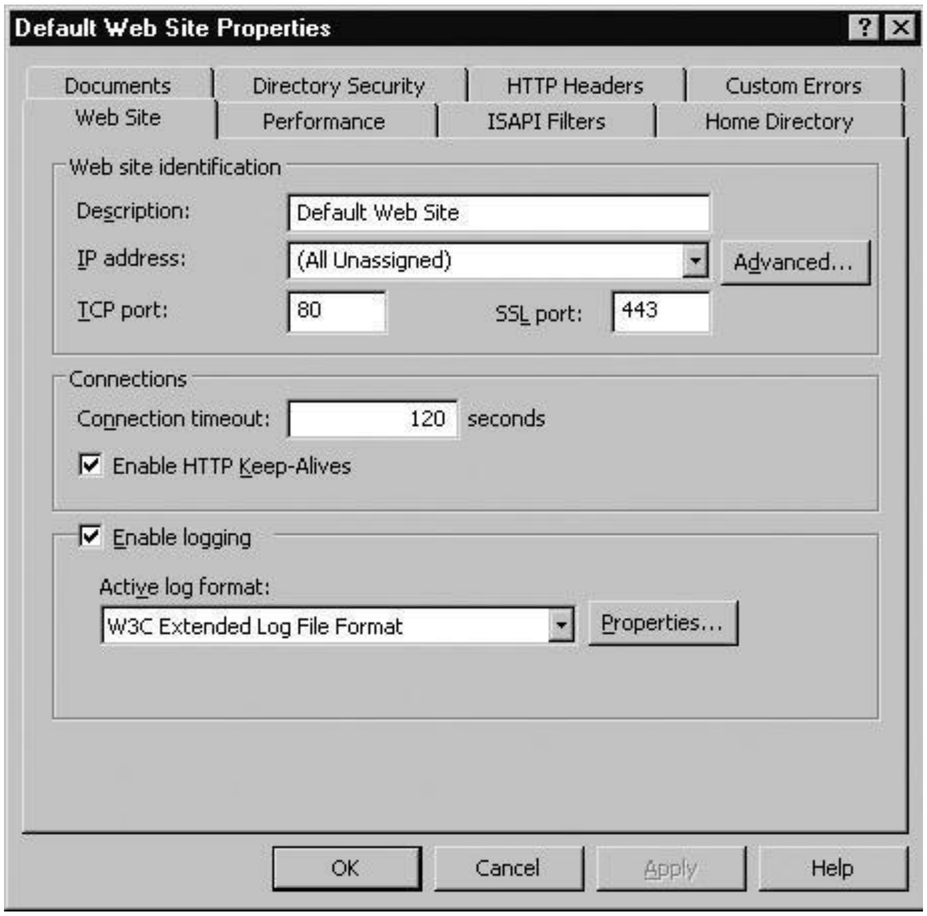
## 7.1 Enabling and Configuring Web Logging

Web logging can be enabled and configured at various levels, and settings at higher levels are inherited by lower levels in the usual way. Web logging can be enabled at two levels:

- **Web Sites level:** Enabling web logging at this level turns it on for all websites on the IIS machine.
- **Individual websites level:** Web logging can also be enabled on a per-website level.

Once web logging has been enabled for a site or for all sites, which specific home directories, virtual directories, subdirectories, and files whose access needs to be logged should be configured. For example, if web administrator wants to log access to only the "default.asp" page within the home directory of the Default Web Site, and not for files in any other sites on the server. The following three steps are needed to make this happen:

1. Open the properties sheet for the Default Web Site and select the Enable Logging check box on the Web Site tab to enable web logging for this particular site (Figure 1). This turns on web logging, but no logs are created unless specific directories and files have been marked for logging within the site (see step 3). However, *all* directories and files are marked by default for logging in IIS, so it's really a matter of specifying which directories and files administrator *don't* want to log access to.



Web site identification
Description: Default Web Site
IP address: (All Unassigned)
TCP port: 80
SSL port: 443

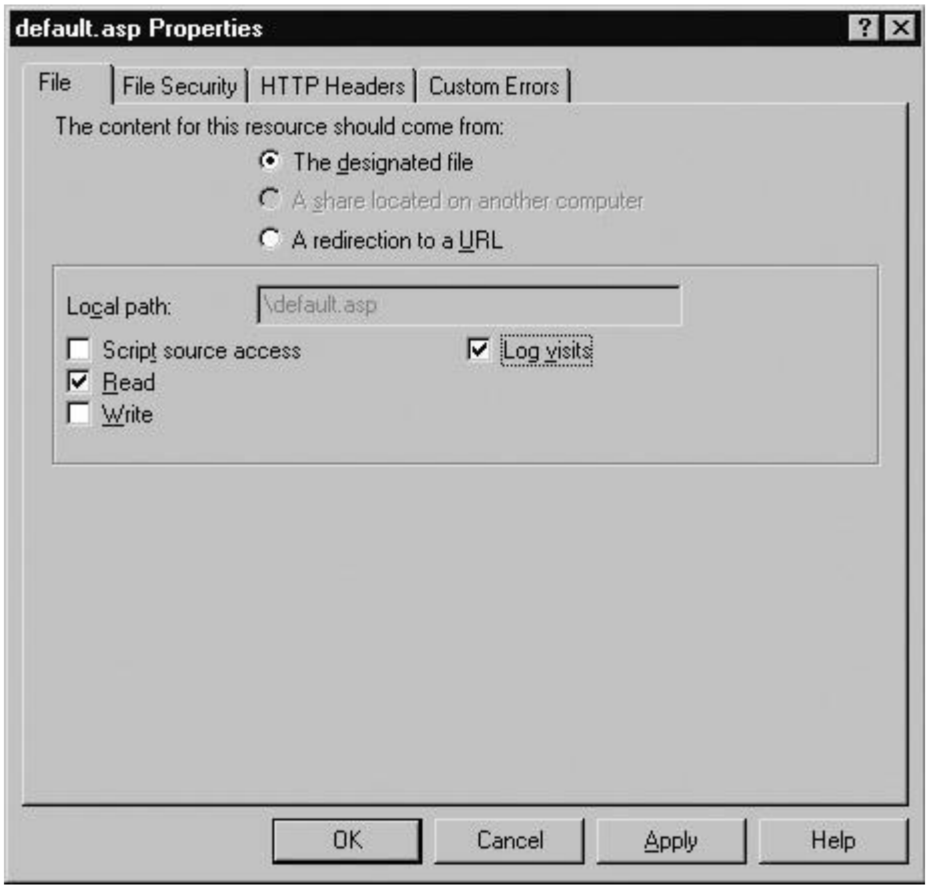
Connections
Connection timeout: 120 seconds
☑ Enable HTTP Keep-Alives

☑ Enable logging
Active log format: W3C Extended Log File Format

Figure-3: Enabling logging for the Default Web Site

2. On the same Web Site tab, select the log file format in which web logs to be recorded using the Active Log Format list box. Click Properties to further configure it by specifying where log files will be stored and various other settings that depend upon the particular format.

3. Finally, mark which directories and files to be logged. If access to everything (physical and virtual subdirectories and files) in website is to be logged, switch to the Home Directory tab and make sure the Log Visits check box is selected. Similarly, to log access to files in a virtual directory, use the Virtual Directory tab on the properties sheet for the virtual directory. To log one specific file, use the File tab on the properties sheet for the file. For example, administrator wants to log access to the file default.asp; so open the properties sheet for the Default Web Site, select the Home Directory tab, clear the Log Visits check box, and click OK. This leaves all files and directories in the site *unmarked* for logging (no logging will occur). Now open the properties sheet for default.asp and select the check box for "Log Visits" on the File tab to mark this file for logging of client connection attempts (Figure 2). Click OK to apply the setting.


default.asp Properties

File | File Security | HTTP Headers | Custom Errors

The content for this resource should come from:

- [✓ The designated file]
- A share located on another computer
- A redirection to a URL

Local path: \default.asp

- [ ] Script source access
- [x] Read
- [ ] Write
- [x] Log visits

Figure-4: Marking the default.asp page for logging

## 7.2 Log File Formats

There are five different logging formats that can be chosen for web logs:

- **W3C Extended Log File format** This is a customizable format developed by the World Wide Web Consortium (W3C) that allows to select which properties to write in logs. By default, IIS selects this format for web logging, and all logging is done in ASCII unless UTF-8 logging [3] is enabled on the IIS machine.
- **IIS Log File format** This is a fixed format that cannot be customized and was developed by Microsoft for early versions of IIS. Although this format records more information than the NCSA format, the IIS log file format is rarely used nowadays, having been superceded by the more powerful and flexible W3C Extended format.

[3] A new feature of version 6 of IIS is support for writing web log files using UTF-8 (a type of Unicode character encoding) instead of the traditional ASCII character encoding that uses the local character set on the machine. The advantage of UTF-8 logging is that data can be written in non-European character sets to the web logs. To enable UTF-8 logging, right-click the *server_name* node in IIS Manager, select Properties, and select the check box labeled Encode Web Logs In UTF-8. Two things to note about UTF-8 logging are that it is a global setting that is turned on or off for all web logging, and it cannot be used for logging visits to FTP sites.

- **NCSA Log File Format** This fixed format cannot be customized and was developed by the National Center for Supercomputing Applications (NCSA) for Mosaic. An advantage of this format is that it is used by the widest range of web server products, but W3C Extended format is more powerful and flexible.
- **ODBC Logging format** This lets IIS log connection attempts to an ODBC- compliant database like Microsoft SQL Server. A major disadvantage of ODBC logging is that enabling it disables kernel-mode caching in IIS, which can significantly degrade IIS server performance. As a result, ODBC logging is not recommended in most circumstances.
- **Centralized Binary Logging format** This is new to version 6 of IIS and allows multiple websites to write to a single log file using a binary format. Centralized binary logging is particularly useful in a web hosting environment where an ISP is hosting hundreds of websites on a single IIS machine and it would consume too much system resources and degrade performance if individual web logs were used for each website.

***Note*** *While all five log file formats can be used for logging visits to websites, only three of them (W3C Extended, ODBC, and IIS formats) can be used for logging visits to FTP sites on IIS.*

In addition to the preceding logging methods included with IIS, web administrator can also develop his own custom logging modules using COM and use them for logging website visits. Unfortunately, using custom logging modules disables kernel-mode caching in IIS and thus degrades performance of web server; so unless there is some compelling reason for using this approach, its best avoided.
Unless web server is working in a web hosting environment where centralized binary logging could be advantageous, the best choice for web logging is W3C Extended format.

## 7.3 Using W3C Extended Logging

To use W3C Extended Logging following steps are needed to be taken.

- Enable logging for the Default Web Site on IIS machine, and use Windows Explorer to navigate to the \Windows\System32\ LogFiles folder where IIS web logs are stored by default. There should be no files present in the folder.
- Now open the URL http://localhost on IIS machine and view the default document for the Default Web Site. In the \LogFiles folder in Windows Explorer, a text file appear with a name something like ex080506.log, where

    - ex stands for W3C Extended format
    - 08 are the last two digits of the current year
    - 05 is the current month
    - 06 is today’s date

- The name of log file will be different. Try opening the file—it will probably be blank because IIS first creates the web log file and then, a short time later, writes the first entry to it, so wait a few seconds and open the file again and something like this can be seen:

```
#Software: Microsoft Internet Information Services 6.0
#Version: 1.0
#Date: 2008-05-06 23:16:46
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-
username c-ip cs(User-Agent) sc-status sc-substatus sc-win32-status
2008-05-06 23:16:46 127.0.0.1 GET /Default.asp—80—127.0.0.1
Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.2;+.NET+CLR+1.1.4322) 200 0 0
```

The first three lines identify the log, while the fourth line (beginning with #Fields) displays the names of the different properties that are logged. The fifth line (beginning with 2008-05-06) displays the log information for the visit that occurred when URL opened http://localhost on the IIS machine. Because the lines are long and wrap in the log text in the text, and because fields are separated by spaces instead of commas or tabs, it's difficult to read and interpret the text. To make it more understandable, refer to following Table-7.

| **Log Field** | **Sample Value** | **Interpretation** |
|---|---|---|
| date | 2008-05-06 | Today |
| time | 23:16:46 | A few minutes ago |
| s-ip | 127.0.0.1 | The Default Web Site on localhost |
| cs-method | GET | HTTP GET request for default document |
| cs-uri-stem | /Default.asp | Default document |
| cs-uri-query | - | No data |
| s-port | 80 | Standard TCP port for HTTP |
| cs-username | - | No data |
| c-ip | 127.0.0.1 | Client browser is also on localhost |
| cs(User-Agent) | Mozilla/4.0+(etc) | Type of web browser used |
| sc-status | 200 | HTTP 200 status code indicates request was successful |
| sc-substatus | 0 | 0 indicates no HTTP substatus code was logged |
| sc-win32-status | 0 | Used by Microsoft Windows |

Table-7: Sample Record for W3C Extended Log File

**Note** A useful property web administrator should always log for troubleshooting purposes is sc-substatus, which logs substatus codes (such as 404.3) for HTTP error messages.

## 7.4 Field Definitions for W3C Extended Logs

To better enable web administrator to read and interpret W3C Extended logs such as the sample file just shown, here is a quick reference to the various fields and what they mean. Note that the different prefixes have specific meanings, for example:

- s- The action occurs on the server
- c- The action occurs on the client
- cs- A client-to-server action (a request)
- sc- A server-to-client action (a response)

Table 8 shows the various fields and their meaning for W3C Extended logging.

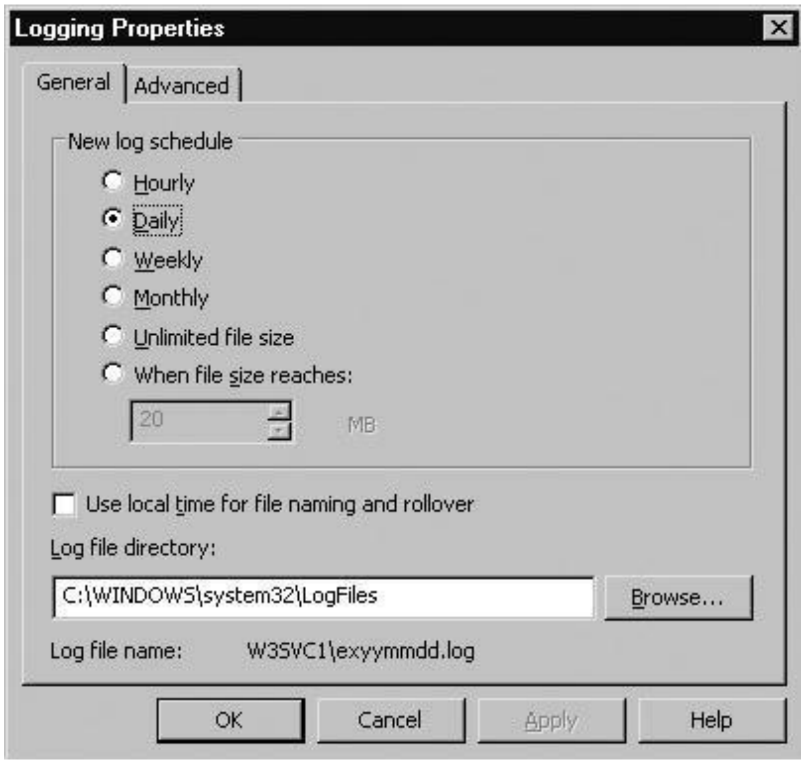
| **Field** | **Appears As** | **Description** |
|---|---|---|
| Date | Date | Date activity occurred |
| Time | Time | Time activity occurred |
| Client IP Address | c-ip | IP address of client accessing server |
| User Name | cs-username | Name of authenticated user (anonymous users indicated by dash) |
| Service Name | s-sitename | Internet service and instance number accessed |
| Server Name | s-computername | Name of server |
| Server IP Address | s-ip | IP address of server |
| Server Port | s-port | Port number used |
| Method | cs-method | HTTP verb used in client request |
| URI Stem | cs-uri-stem | File accessed |
| URI Query | cs-uri-query | Query performed by client |
| Protocol Status | sc-status | Status of action in HTTP or FTP terms |
| Win32® Status | sc-win32-status | Status of the action in terms used by Microsoft Windows |
| Bytes Sent | sc-bytes | Number of bytes sent by server to client |
| Bytes Received | cs-bytes | Number of bytes received by server from client |
| Time Taken | time-taken | Time duration (milliseconds) |

| Field | Appears As | Description |
|---|---|---|
| | | consumed by action |
| Protocol Version | cs-version | Protocol version used by client |
| Host | cs-host | Contents of host header |
| User Agent | cs(User-Agent) | Web browser used by client |
| Cookie | cs(Cookie) | Content of cookie sent or received (if any) |
| Referrer | cs(Referer) | Last site visited by user if redirected |

Table 8: Summary of Field Definitions for W3C Extended Log File Format

## 7.5 Scheduling W3C Extended Logging

To configure when W3C web logs will be created and what information will be written to them, click Properties on the Web Sites tab of website to open Logging Properties. The General tab on this properties sheet (Figure 5) is used for configuring the following aspects of logging:



New log schedule: [✓ Daily]

20 MB

☐ Use local time for file naming and rollover

Log file directory: C:\WINDOWS\system32\LogFiles

Log file name: W3SVC1\exyymmdd.log

Figure-5: Specifying a new log schedule and log file directory

- **New Log Schedule** Indicates the condition under which new log files are created. This can be hourly, daily, weekly, monthly, or when the current log reaches a specified size, or keep on logging to the same file and let it grow as big as web administrator wants. Daily is the default option, and it creates a new log file when the first visit occurs after midnight. On high-volume sites, web administrator might choose the Hourly option, but don't choose Unlimited File Size unless there is terabytes of disk space available on IIS machine! Log files are named differently depending on the new log schedule selected, and these naming conventions are summarized in Table 9.

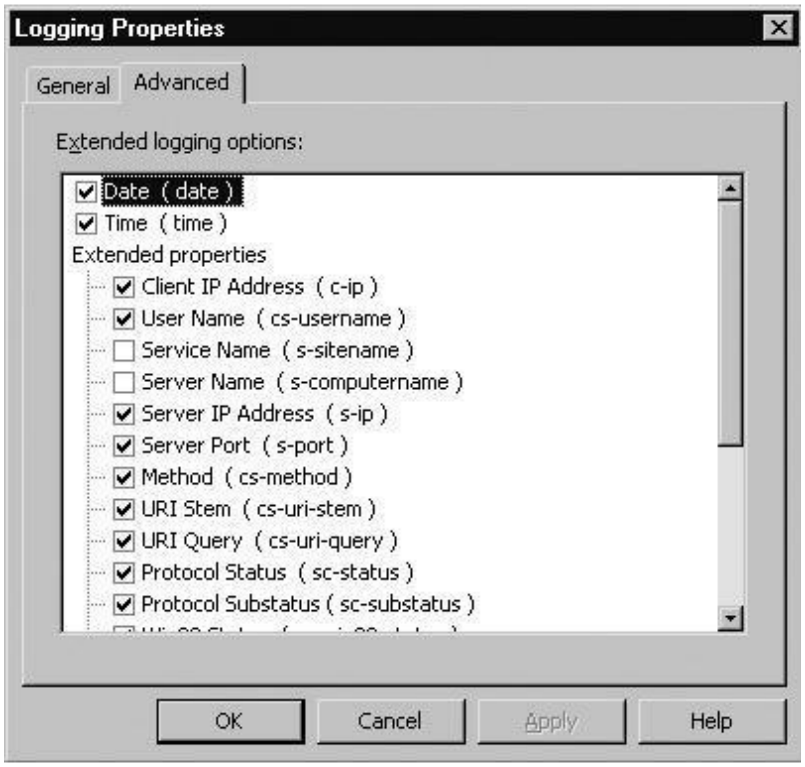
| New Log Schedule | Pattern for Log Filename |
|---|---|
| Hourly | exyymmddhh.log |
| Daily | exyymmdd.log |
| Weekly | exyymmww.log |
| Monthly | exyymm.log |
| Unlimited | extend#.log |
| Specified size | extend#.log |

Table 9: Naming Conventions for W3C Extended Log Files

- **Use Local Time for File Naming and Rollover** Note that the Daily option creates new log files at midnight, but for W3C Extended logging this means midnight Coordinated Universal Time (UTC), which is essentially midnight Greenwich Mean Time (GMT). Web administrator can cause IIS to create new log files at midnight local time by selecting this option, but even if he does so the visit times recorded in the log file will be UTC times. Then web administrator has to convert the times in log files to local time zone.
- **Log File Directory** The default location for saving web logs is within a subdirectory of the \Windows\System32\LogFiles folder. The name of the subdirectory depends on the log file format chosen, and for W3C Extended format the subfolder is \W3SVCsite_id, where site_id is the website identifier displayed for the site when the Web Sites node is selected in IIS Manager. The Default Web Site is assigned the identifier 1 (one) so the actual location where W3C Extended log files will be stored for the Default Web Site is the \Windows\System32\LogFiles\W3SVC1 folder.

## 7.6 Configuring W3C Extended Logging

In addition to scheduling when new log files will be created and where they will be saved, web administrator can also configure what properties are saved in these files for each record generated by a client connection attempt. To configure which properties to log, use the Advanced tab on Logging Properties (Figure 6).


Logging Properties

Extended logging options:

- [✓] Date ( date )
- [✓] Time ( time )

Extended properties

- [✓] Client IP Address ( c-ip )
- [✓] User Name ( cs-username )
- [ ] Service Name ( s-sitename )
- [ ] Server Name ( s-computername )
- [✓] Server IP Address ( s-ip )
- [✓] Server Port ( s-port )
- [✓] Method ( cs-method )
- [✓] URI Stem ( cs-uri-stem )
- [✓] URI Query ( cs-uri-query )
- [✓] Protocol Status ( sc-status )
- [✓] Protocol Substatus ( sc-substatus )

Figure-6: Specifying which properties to log

**Note** Don’t log too many properties in web logs. Not only do that need adequate disk space for logs, but too much logging also adds a performance hit to IIS by consuming memory and processor resources. Log only what is needed, and only for the sites or portions of sites (virtual directories, physical directories, and files) that are most concerned about for reasons of security or performance or for gathering statistics..

## 7.7 Securing Web Logs

Web administrator can better protect web logs from intruders by moving them to a different directory than the default directory IIS uses to store them, which is the \Windows\System32\LogFiles folder. To change the location of W3C Extended, IIS, or NCSA log files, click the Properties button on the Web Site tab of site’s properties sheet and specify a new Log File Directory in the text box (refer to Figure 3).
To secure new log file folder, make sure to assign appropriate ACLs to it as follows:

- Administrators Full Control
- IIS_WPG group Full Control
- SYSTEM Full Control

## 7.8 Remote Logging

While earlier versions of IIS required the log file directory be located on the local machine, version 6 lets log to a shared folder on a file server on network. To do this, type the UNC path to the share in the Log File Directory textbox (shown in Figure 3). This new approach has the advantage of allowing centralizing the backup of log files from multiple IIS machines to a single file server, but because of the performance penalty incurred, it is generally not recommended. If web administrator does decide to employ remote logging for IIS machines, Microsoft suggests using IPSec for the connection between web and file server should be considered, as anyone sniffing the network can read log files because they are being sent as plain text.

## 7.9 Managing Web Logs

To delete web logs, first stop the website the logs are associated with, and then delete them using Windows Explorer. If the website is not stopped, the log files may not be able to delete due to a sharing violation.

To review web logs like W3C Extended logs that are written in ASCII format, open them using a text editor like Notepad or import them into a spreadsheet or database program for analysis. If selected the Hourly or Daily new log option, there would be too many log files to review, use the copy command to combine multiple text files into a single large file. Type "copy /?" at the command line for information on how to append multiple files.

**Note** If IIS runs out of disk space and can't add records to the log file, IIS will automatically shut down and an event will be recorded in the Application log of Event Viewer. When more disk space is added (for example, by extending the volume), IIS starts and logging resumes.

### Analysis and Reporting Tools

Analysis of Logs collected from Webserver is next important task. Notepad is useful for a quick peek at log files that have small amounts of information in them, but web servers running in service provider or e-commerce environments need something more powerful for analyzing and generating reports from IIS logs. Microsoft has provided Log Parser 2.0 (free tool) to perform SQL-like queries on IIS log files and display the results.

## 7.10 Converting Log Formats

The different web log formats supported by IIS are not compatible with one another. For example, while both NCSA and W3C Extended logging record dates using four- digit year format, IIS logging records dates earlier than 1999 as two-digit years, and four-digit years thereafter. Furthermore, IIS log file format separates fields using commas, but NCSA and W3C Extended formats use spaces as separators instead. Another difference is that while W3C Extended records event times using UTC (coordinated universal time, sometimes called Greenwich Mean Time or GMT), the IIS and NCSA formats employ local time instead.

To compare web logs generated using different log file formats, Windows Server 2003 includes the utility convlog.exe in \Windows\System32 that can be used to convert log files that use W3C Extended or IIS log file formats to standard NCSA format. The conversion process automatically performs a time zone offset (using an offset specified by command user) in order to change the times in the log from UTC to local time, making them easier to read and interpret. In addition, this utility can be used to replace IP addresses in log files with DNS domain names associated with them. For syntax on how to use this tool, type “convlog /?” at the command line or see IIS Help.

## 7.11 IIS and Event Logs

IIS also generates error, warning, and informational events that can be displayed using Event Viewer. Whenever a problem occurs with IIS, it’s a good idea to review the System and Application logs in Event Viewer to see if there is any useful information there that can be used for troubleshooting purposes. Events can be logged for IIS services like the WWW Service or FTP Service, for ASP or ASP.NET applications, and so on.

The list of possible events IIS can generate is long; for a detailed list, see the Events Reference in IIS Help.

# 8. Auditing and Logging for Apache Web Server

## 8.1 Audit Configuration for Apache Web Server

The Apache auditing mechanism allows an administrator to record the activities on the Apache server. By observing the log files, administrators can determine site traffic, requested resources, resource retrieval method and whether or not the server experienced difficulties serving the request. It can help determine whether a client is attempting to perform illicit activities on the server and can give some insights as to the nature of an attack if a violation occurs. For this reason, the auditing mechanism is of great importance to the security of the server.

## 8.2 Audit Modules

A typical Apache Web Server contains the following modules which relate to recording usage information. The modules are:
***core*** —The main Apache module. This contain directives concerning the error log which records problems and events experienced by the Apache server independent of requests made by clients.

***mod_log_config***—The primary auditing module. This module allows an administrator to create audit facilities which will record wide range of information concerning client requests.

***mod_log_agent***—Use to record information about the nature of a client making a request.

***mod_log_referer***—Used to record information concerning what resource provided the link to the resource being requested.

***mod_usertrack***—Allows the administrator to track the activities of a client when accessing the server.

## 8.3 Default Configuration

By default, the *mod_log_config* module is enabled. The default Apache configurations files contain several active directives from both the core and *mod_log_config* modules. It also contains several directives that have been commented out, but which can be uncommented for swift implementation. The pertinent directives in the default Apache configuration files are provided as an example below. It is described in detail in following section 8.4

```
ErrorLog /usr/local/apache/var/log/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\"
\"%{User-Agent}i\"" combined
<ip of client> <remote logname> <username> <access time>
"<first line of request>" <final status> <bytes sent>
"<Referer field>" "<User-Agent field>"
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
<ip of client> <remote logname> <username> <access time>
"<first line of request>" <final status> <bytes sent>
LogFormat "%{Referer}i -> %U" referrer
<Refer field> -> <URL requested>
LogFormat "%{User-agent}i" agent
<User-agent field>
CustomLog /usr/local/apache/var/log/access_log common
```

## 8.4 An example Configuration File for Logging

```
ErrorLog /usr/local/apache/var/log/error_log
```

This line sets the location and name of the server error log.

```
LogLevel warn
```

This line sets the error log to report incidents of warn priority or higher. Refer Section 8.6 for more concerning what events each level record.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\"
\"%{User-Agent}i\"" combined
```

This line defines a log format and gives it the nickname “combined” for later use.

This format definition would produce an event with the following form and information:

```
<ip of client> <remote logname> <username> <access time>
"<first line ofrequest>" <final status> <bytes sent>
"<Referer field>" "<User-Agentfield>"

LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

This line defines a log format and gives it the nickname “common” for later use. (This format represents Common Log Format.) This format definition would produce an event with the following form and information:

```
<ip of client> <remote logname> <username> <access time>
"<first line of request>" <final status> <bytes sent>

LogFormat "%{Referer}i -> %U" referrer
```

This line defines a log format and gives it the nickname “referer” for later use. (The resulting format is the same as that provided by the mod_log_referer module.) This format displays the following fields on each line:

```
<Refer field> -> <URL requested>
```

```
    LogFormat "%{User-agent}i" agent
```

This line defines a log format and gives it the nickname "agent" for later use. (The resulting format is virtually the same as that provided by the *mod_log_agent* module.)

This format displays the following field on each line:

```
    <User-agent field>
```

```
    CustomLog /usr/local/apache/var/log/access_log common
```

This line creates a new log file with the format associated with the "common" nickname. The new log file is named "*access_log*" and is located in the `/usr/local/apache/var/log/` directory.

```
    #CustomLog /usr/local/apache/var/log/referer_log referrer
```

This line creates a new log file with the format associated with the "referer" nickname. The new log file is named "referer_log" and is located in the `/usr/local/apache/var/log/` directory. The line is commented out so, by default, this log file is not created.

```
    #CustomLog /usr/local/apache/var/log/agent_log agent
```

This line creates a new log file with the format associated with the "agent" nickname. The new log file is named "agent_log" and is located in the `/usr/local/apache/var/log/` directory. The line is commented out so, by default, this log file is not created.

```
    #CustomLog /usr/local/apache/var/log/access_log combined
```

This line creates a new log file with the format associated with the "combined" nickname. The new log file is named "access_log" and is located in the `/usr/local/apache/var/log/` directory. The line is commented out so, by default, this log file is not created.

## 8.5 Important Logs in Apache Server

By default, the Apache server creates two log files: The error log and the access log. The error log, named ***error_log***, is part of the Apache ***core*** module and records events on the server independent of client requests. The access log, named ***access_log***, records information about user requests in Common Log Format.

## 8.6 Error Logs

The server error log, part of the Apache ***core*** module, whose name and location is set by the **`ErrorLog`** directive, is the most important log file. This is the place where Apache ***httpd*** will send diagnostic information and record any errors that it encounters in processing requests. It is

the first place to look when a problem occurs with starting the server or with the operation of the server, since it will often contain details of what went wrong and how to fix it.
Location of the error log

**ErrorLog** `logs/error_log`

[It may be noted that path is Relative to ServerRoot]

The above configuration will work for logging all error-related information in one error log file, even multiple domains are hosted on a server.

The **`LogLevel`** directive is used to control the types of errors that are sent to the error log by restricting the severity level. This adjusts the verbosity of the messages recorded in the error logs. Some example errors and corresponding level are shown in Table 10.

| Level | Description | Example |
|---|---|---|
| **`emerg`** | Emergencies - system is unusable. | "Child cannot open lock file. Exiting" |
| **`alert`** | Action must be taken immediately. | "getpwuid: couldn't determine user name from uid" |
| **`crit`** | Critical Conditions. | "socket: Failed to get a socket, exiting child" |
| **`error`** | Error conditions. | "Premature end of script headers" |
| **`warn`** | Warning conditions. | "child process 1234 did not exit, sending another SIGHUP" |
| **`notice`** | Normal but significant condition. | "httpd: caught SIGBUS, attempting to dump core in ..." |
| **`info`** | Informative. | "Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)..." |
| **`debug`** | Debug-level messages | "Opening config file ..." |

Table 10: Log Levels and errors

When a particular level is specified, messages from all other levels of higher significance will be reported as well. For example, when **`LogLevel info`** is specified and then messages with log levels of **`notice`** and **`warn`** will also be posted.

Using a level of at least **`crit`** is recommended.

It is not possible to customize the error log by adding or removing information. However, error log entries dealing with particular requests have corresponding entries in the access log.

## 8.7 Typical errors

The typical errors logged by access log are explained below:

- Document error

```
[Fri Aug 18 22:36:26 2008] [error] [client 192.168.1.6] File does not exist:
/usr/local/apache/bugletdocs/Img/south-korea.gif
```

As with access log, error message is in several distinct parts

- Authentication error

```
[Tue Apr 11 22:13:21 2008] [error] [client 192.168.1.3] user rbowen
authentication failure for "/cgi-bin/hirecareers/company.cgi":password mismatch
```

- CGI errors

```
[Wed Jun 14 16:16:37 2008] [error] [client 192.168.1.3] Premature end of script headers:
/usr/local/apache/cgi-bin/TestProg/announcement.cgi
Global symbol "$rv" requires explicit package name at
/usr/local/apache/cgi-bin/TestProg/announcement.cgi line 81.
Global symbol "%details" requires explicit package name at
/usr/local/apache/cgi-bin/TestProg/announcement.cgi line 84.
Global symbol "$Config" requires explicit package name at
/usr/local/apache/cgi-bin/TestProg/announcement.cgi line 133.
Execution of /usr/local/apache/cgi-bin/TestProg/announcement.cgi
```

## 8.8 Access Logs

The server access log records all requests processed by the server. It is a part of ***mod_log_config*** module. This module provides for flexible logging of client requests. Logs are written in a customizable format, and may be written directly to a file, or to an external program.

Three directives are provided by this module: **`TransferLog`** to create a log file, **`LogFormat`** to set a custom format, and **`CustomLog`** to define a log file and format in one step. Conditional logging is provided so that individual requests may be included or excluded from the logs based on characteristics of the request.

There are different ways to create audit files using these directives. Default Apache configuration file use **`LogFormat`** to set a custom format and **`CustomLog`** to define a log file. The other two ways are **`LogFormat`** can describe a format and **`TransferLog`** can create the file and **`CustomLog`** can define the format and create the file.

The **`TransferLog and CustomLog`** directives can be used multiple times in each server to cause each request to be logged to multiple files.

## 8.9 Log Formats

By default Apache uses the Common log format. It is recommended that to use Combined Log Format because the Common Log Format does not provide information about referrers and User agents (OS, Browsers etc). Log format can be configured by editing the "httpd.conf" file in the Apache conf default directory [/etc/httpd/conf/httpd.conf].

A typical configuration is as below:

```
LogFormat "%h %l %u %t \"%r\" %<s %b" common
CustomLog logs/access_log common
```

The format of the access log is highly configurable. The format argument to the **`LogFormat`** and **`CustomLog`** directives is a string. This string is used to log each request to the log file. It can contain literal characters copied into the log files and the C-style control characters "\n" and "\t" to represent new-lines and tabs. Literal quotes and back-slashes should be escaped with back-slashes.

The characteristics of the request itself are logged by placing "%" directives in the format string, which are replaced in the log entry by the values as follows:

```
%...a:          Remote IP-address
%...A:          Local IP-address
%...B:          Bytes sent, excluding HTTP headers.
%...b:          Bytes sent, excluding HTTP headers. In CLF format
                 i.e. a '-' rather than a 0 when no bytes are sent.
%...c:          Connection status when response was completed.
                'X' = connection aborted before the response completed.
                '+' = connection may be kept alive after the response is
                      sent.
                '-' = connection will be closed after the response is sent.
%...{FOOBAR}e:  The contents of the environment variable FOOBAR
%...f:          Filename
%...h:          Remote host
%...H:          The request protocol
%...{Foobar}i:  The contents of Foobar: header line(s) in the request
                sent to the server.
%...l:          Remote logname (from identd, if supplied)
%...m            The request method
%...{Foobar}n:  The contents of note "Foobar" from another module.
%...{Foobar}o:  The contents of Foobar: header line(s) in the reply.
%...p:          The canonical Port of the server serving the request
%...P:          The process ID of the child that serviced the request.
%...q            The query string (prepended with a ? if a query string
                 exists, otherwise an empty string)
%...r:          First line of request
```

```
%...s:          Status.  For requests that got internally redirected, this is
                the status of the *original* request --- %...>s for the last.
%...t:          Time, in common log format time format (standard English
                 format)
%...{format}t:  The time, in the form given by format, which should
                be in strftime(3) format. (potentially localized)
%...T:          The time taken to serve the request, in seconds.
%...u:          Remote user (from auth; may be bogus if return status (%s) is
                 401)
%...U:          The URL path requested, not including any query string.
%...v:          The canonical ServerName of the server serving the request.
%...V:          The server name according to the UseCanonicalName setting.
```

### 8.9.1 Some commonly used log format strings are:

Common Log Format (CLF)

```
"%h %l %u %t \"%r\" %>s %b"
```

Common Log Format with Virtual Host

```
"%v %h %l %u %t \"%r\" %>s %b"
```

NCSA extended/combined log format

```
"%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
agent}i\""
```

Referer log format

```
"%{Referer}i -> %U"
```

Agent (Browser) log format

```
"%{User-agent}i"
```

### 8.9.2 Common Log Format

Log Format "%h %l %u %t \"%r\" %>s %b"

**Example :**

```
127.0.0.1 - frank [10/Oct/2008:13:55:36 -0700] "GET
/apache_pb.gif HTTP/1.0" 200 2326
```

Each part of this log entry is described below.

**`127.0.0.1`**
This is the IP address of the client (remote host) which made the request to the server. If "Hostnamelookups" is set to On, then the server will try to determine the hostname and log it in place of the IP address.The IP address reported here is not necessarily the address of the machine at which the user is sitting. If a proxy server exists between the user and the server, this address will be the address of the proxy, rather than the originating machine.

**-**
The "hyphen" in the output indicates that the requested piece of information is not available. In this case, the information that is not available is the RFC 1413 identity of the client determined by ***identd*** on the clients machine.

**`frank`**
This is the “user id” of the person requesting the document as determined by HTTP authentication. If the status code for the request (see below) is 401, then this value should not be trusted because the user is not yet authenticated. If the document is not password protected, this entry will be "-" just like the previous one.

**`[10/Oct/2008:13:55:36 -0700]`**

The time that the server finished processing the request. The format is:

**`[day/month/year:hour:minute:second zone]`**

**`"GET /apache_pb.gif HTTP/1.0"`**
The request line from the client is given in double quotes. The request line contains a great deal of useful information. First, the method used by the client is GET. Second, the client requested the resource /apache_pb.gif, and third, the client used the protocol HTTP/1.0.

**`200`**
This is the status code that the server sends back to the client. This information is very valuable, because it reveals whether the request resulted in a successful response (codes beginning in 2), a redirection (codes beginning in 3), an error caused by the client (codes beginning in 4), or an error in the server (codes beginning in 5).

**`2326`**
The last entry indicates the size of the object returned to the client, not including the response headers. If no content was returned to the client, this value will be "-".

### 8.9.3 Combined Log Format

This format is exactly the same as the Common Log Format, with the addition of two more fields.

```
LogFormat "%h %l %u %t \"%r\" %>s %b\"%{Referer}i\" \"%{User-Agent}i\""
```

**Example:**

```
127.0.0.1 - frank [10/Oct/2008:13:55:36 -0700] "GET
/apache_pb.gif HTTP/1.0" 200 2326
"http://www.example.com/start.html" "Mozilla/4.08 [en]
(Win98; I ;Nav)"
```

The additional fields are:

**`"http://www.example.com/start.html"`**

The "Referer" (sic) HTTP request header. This gives the site that the client reports having been referred from. (This should be the page that links to or includes /apache_pb.gif).

**`"Mozilla/4.08 [en] (Win98; I ;Nav)"`**

The User-Agent HTTP request header; This is the identifying information that the client browser reports about itself.

## 8.10 Securing Log Files

All log files created by the Apache server are owned by the “root user” and are only writable by the “root user” and readable by all users. The default Apache installation places log files in a directory which all users can enter and browse; but only the “root identity” can add files to this directory. For the most part, this is considered a secure setup; it prevents anyone but “root” from changing or replacing log files without first defeating the security of the Linux operating system. Many administrators find that letting users read the log files is more access than they can allow. Log files can sometimes contain sensitive information including, server vulnerability alerts, the true paths of resources, or simply information about server usage. For these reasons, many administrators choose to prevent read access to all but themselves.

This can be accomplished in several ways. The easiest is to modify the security of the directory into which the logs are written, making the directory owned by the server administrator. The administrator has full access to the directory while all other users are given no access. Since events are logged under the root user identity, the Apache server will always be able to record events no matter what the security settings are on the directory.
For example, if ServerRoot (The ServerRoot directive sets the directory in which the server lives) is to be placed in /usr/local/apache (default path) then it is suggested that the directory may be created as root, with commands like these:

```
mkdir /usr/local/apache

cd /usr/local/apache

 mkdir bin conf logs

chown 0 . bin conf logs

chgrp 0 . bin conf logs

chmod 755 . bin conf logs
```

It is assumed that /, /usr, and /usr/local are only modifiable by "root". While installing the ***httpd*** executable, ensure that it is similarly protected:

```
cp httpd /usr/local/apache/bin

chown 0 /usr/local/apache/bin/httpd

chgrp 0 /usr/local/apache/bin/httpd

chmod 511 /usr/local/apache/bin/httpd
```

## 8.11 Managing Logs

### 8.11.1 Sending Apache Logs to the syslog Mechanism

Many services on Linux send their log events to the ***syslog*** mechanism. The Apache server can also pipe the events which would normally be sent to the ***error log*** to the ***syslog*** mechanism to be placed with the events from the other services.

### 8.11.2 Log Rotation

On even a moderately busy server, the quantity of information stored in the log files is very large. The access log file typically grows 1 MB or more per 10,000 requests. It will consequently be necessary to periodically rotate the log files by moving or deleting the existing logs. This cannot be done while the server is running, because Apache will continue writing to the old log file as long as it holds the file open. Instead, the server must be restarted after the log files are moved or deleted so that it will open new log files.

By using a graceful restart, the server can be instructed to open new log files without losing any existing or pending connections from clients. However, in order to accomplish this, the server must continue to write to the old log files while it finishes serving old requests. It is therefore necessary to wait for some time after the restart before doing any processing on the log files. A typical scenario that simply rotates the logs and compresses the old logs to save space is:

```
mv access_log access_log.old
mv error_log error_log.old
apachectl graceful
sleep 600
gzip access_log.old error_log.old
```

**Piped Logs**

Another way to perform log rotation is using "piped logs". One important use of piped logs is to allow log rotation without having to restart the server. The Apache HTTP Server includes a simple program called *"rotatelogs"* for this purpose. For example, to rotate the logs every 24 hours, we can use:

```
CustomLog "|/usr/local/apache/bin/rotatelogs
/var/log/access_log 86400" common
```

# 9. Auditing for Database Server – Oracle 10G

## 9.1 Introduction

The Oracle Database Server provides a fairly robust set of auditing capabilities. This section introduces the basic commands of auditing an Oracle database. Various auditing alternatives are available in Oracle's RDBMS. This section describes how to conduct an audit in oracle, why and when audit is required in oracle database. Different aspects of Oracle auditing are illustrated with examples in this guideline.

## 9.2 Auditing of Users

Oracle’s first form of auditing is a subsystem which can be used to record failed and successful attempts on the server. Recording connection attempts is useful in being able to discover:

- Who is attempting to connect to the database?
- When an attack has occurred?
- Whether the attack was successful or not?

The auditing is to be enabled to capture user access, use of system privileges and changes to the database schema structure. This basic set will not show attempted access to specific data that shouldn't be accessed; however, it will give a reasonably simple overview of "incorrect" access and use of privileges. If a user is suspected of inappropriate actions or if an attack has been suspected then more detailed audit can be turned on for specific tables.

The standard audit commands allow all system privileges to be audited along with access at the object level to any table or view on the database for select, delete, insert or update. Audit can be performed for either successful or unsuccessful attempts or both. It can be for each individual user or for all users and it can also be done at the session level or access level. At action level a single record is created per action and at session level one record is created for all audit actions per session.

## 9.3 Performance and Complexity of Audit

Auditing of database is generally complex and slow. If many or all options are turned on, then the resultant audit trail produced can be large and difficult to interpret and manage. Furthermore, if audit is used on all tables and views in the database, then this can have an effect on performance. Every time an auditable action is performed, a record is written to the database; clearly the more audit is used, the more records will be written to the system table space purely for audit.

So use only the audit that is needed to give an overall view of what is happening and for detailed monitoring of critical data and objects. The simpler the audit trail set-up, it is more likely that the data will be analyzed and be of some use. It is important to define what actions or abuses are being checked for; so that simple reports can be written to filter the audit trail for those actions. With default installation of Oracle, the audit is turned off

It should be noted that the standard audit commands do not allow audit to be performed at row level.

## 9.4 Methods to audit an Oracle database

### 9.4.1 Oracle audit

The database has the ability to audit all actions that take place within it. Audit records may be written to either the SYS.AUD$ table or the operating system's audit trail.

Following three types of actions may be audited

- Login attempts
- Object accesses
- Database actions

All privileges that can be granted to a user or role within the database can be audited. This includes read, write and delete access on objects at the table level. For more detailed audit, the database triggers need to be employed.

### 9.4.2 System triggers

These were introduced with Oracle 8 and allow the writing of database triggers that fire when system events take place. These include start-up and shutdown of the database, log-on and log-off attempts, creation, altering and dropping of schema objects. With the aid of autonomous transactions, these allow a log to be written for the above-mentioned system events.

### 9.4.3 Update, Delete, and Insert triggers

This is the second line of defense in trying to understand users' actions at a more detailed row level. Database triggers need to be written to capture changes at the column and row level. It is possible to write complete rows of data before and after the change being made to a log table in the database. The use of this type of logging is very resource intensive, as many extra records are written and stored. The one failing with this method is that read access cannot be captured with normal database triggers.

### 9.4.4 Fine-grained audit

Fine-grained auditing solves the problem of capturing read access. This feature is based on internal triggers that fire when any piece of SQL is parsed. This is very efficient, as the SQL is parsed once for audit and execution. The feature uses predicates[4] that are defined and tested each time the relevant object is accessed. Fine-grained audit is managed by a PL/SQL package called DBMS_FGA. A PL/SQL procedure is executed every time a "match" is made with the predicate. This method allows the audit to be performed down to the row and column level and to also for read statements. To use of this feature requires programming skills.

### 9.4.5 System logs

Oracle generates many log files and many of them can provide useful information to assist in auditing the database. One good example is the alert log used by the database to record start-up and shutdown as well as any structural changes such as adding a datafile to the Oracle database.

## 9.5 Setting up Auditing

### 9.5.1 Server Setup

Auditing is an in-built feature of the Oracle server. The initialization parameters that influence its behaviour can be displayed using the SHOW PARAMETER SQL*Plus command (refer the command below)

```
SQL> SHOW PARAMETER AUDIT

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------
audit_file_dest                      string
C:\ORACLE\PRODUCT\10.2.0\ADMIN
                                                 \DB10G\ADUMP
audit_sys_operations                 boolean     FALSE
audit_trail                          string      NONE

SQL>
```

Auditing is disabled by default, but can be enabled by setting the AUDIT_TRAIL static parameter, which has the following allowed values.

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

The following list provides a description of each setting:

none or false - Auditing is disabled.

db or true - Auditing is enabled, with all audit records stored in the database audit trial (SYS.AUD$).

[4] Predicates allow the searching through database records to recover specific strings and ranges or characters.

db,extended - As db, but the SQL_BIND and SQL_TEXT columns are also populated.

xml- Auditing is enabled, with all audit records stored as XML format OS files.

xml,extended - As xml, but the SQL_BIND and SQL_TEXT columns are also populated.

os- Auditing is enabled, with all audit records directed to the operating system's audit trail.

The AUDIT_SYS_OPERATIONS static parameter enables or disables the auditing of operations issued by users connecting with SYSDBA or SYSOPER privileges, including the SYS user. All audit records are written to the OS audit trail.

The AUDIT_FILE_DEST parameter specifies the OS directory used for the audit trail when the os, xml and xml,extended options are used. It is also the location for all mandatory auditing specified by the AUDIT_SYS_OPERATIONS parameter.

### Commands to enable auditing and direct audit records to the database audit trail

```
SQL> ALTER SYSTEM SET audit_trail=db SCOPE=SPFILE;

System altered.

SQL> SHUTDOWN
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> STARTUP
ORACLE instance started.
Total System Global Area 360406956 bytes
Fixed Size                  1348400 bytes
Variable Size              61303856 bytes
Database Buffers           24150508 bytes
Redo Buffers                3045024 bytes
Database mounted.
Database opened.

SQL>
```

## 9.5.2 Audit Options

The AUDIT command syntax will give an idea of flexibility in Oracle auditing. While using it the information will not be repeated again and again, for example,

**1. Let the User be AUDIT_CERT-IN_DATA**.

```
CREATE USER AUDIT_CERT-In_DATA IDENTIFIED BY password
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp
  QUOTA UNLIMITED ON users;

GRANT connect TO AUDIT_CERT-In_DATA;
GRANT create table, create procedure TO AUDIT_CERT-In_DATA;
```

CONNECT sys/password AS SYSDBA

**Next, audit all operations by the AUDIT_CERT-IN_DATA user.**

```
CONNECT sys/password AS SYSDBA
AUDIT ALL BY AUDIT_CERT-In_DATA BY ACCESS;
AUDIT SELECT TABLE, UPDATE TABLE, INSERT TABLE, DELETE TABLE BY
AUDIT_CERT-In_DATA BY ACCESS;
AUDIT EXECUTE PROCEDURE BY AUDIT_CERT-In_DATA BY ACCESS;
```

**These options audit all DDL and DML, along with some system events.**

- DDL (CREATE, ALTER & DROP of objects)
- DML (INSERT UPDATE, DELETE, SELECT, EXECUTE).
- SYSTEM EVENTS (LOGON, LOGOFF etc.)

**Next, perform some operations that will be audited.**

```
CONN AUDIT_CERT-In_DATA/password
CREATE TABLE test_tab (
  id  NUMBER
);

INSERT INTO test_tab (id) VALUES (1);
UPDATE test_tab SET id = id;
SELECT * FROM test_tab;
DELETE FROM test_tab;

DROP TABLE test_tab;
```

### 9.5.3 View Audit Trail

The database has the ability to audit all actions that has taken place within it. Audit records may be written to either the SYS.AUD$ table for the operating system's audit trail which is operating system dependent.
Its contents can be viewed directly or via the following views:

```
SELECT view_name
FROM   dba_views
WHERE  view_name LIKE 'DBA%AUDIT%'
ORDER BY view_name;

VIEW_NAME
------------------------------
DBA_AUDIT_EXISTS
DBA_AUDIT_OBJECT
DBA_AUDIT_POLICIES
DBA_AUDIT_POLICY_COLUMNS
DBA_AUDIT_SESSION
DBA_AUDIT_STATEMENT
DBA_AUDIT_TRAIL
DBA_COMMON_AUDIT_TRAIL
DBA_FGA_AUDIT_TRAIL
DBA_OBJ_AUDIT_OPTS
DBA_PRIV_AUDIT_OPTS
DBA_REPAUDIT_ATTRIBUTE
DBA_REPAUDIT_COLUMN
DBA_STMT_AUDIT_OPTS

14 rows selected.
```

The three main views are:

- DBA_AUDIT_TRAIL - Standard auditing only (from AUD$)
- DBA_FGA_AUDIT_TRAIL - Fine-grained auditing only (from FGA_LOG$)
- DBA_COMMON_AUDIT_TRAIL - Both standard and fine-grained auditing

The most basic view of the database audit trail is provided by the DBA_AUDIT_TRAIL view, which contains a wide variety of information. The following query displays some of the information from the database audit trail.

```
COLUMN username FORMAT A10
COLUMN owner    FORMAT A10
COLUMN obj_name FORMAT A10
COLUMN extended_timestamp FORMAT A35

SELECT username,
       extended_timestamp,
       owner,
       obj_name,
       action_name
FROM   dba_audit_trail
WHERE  owner = 'AUDIT_CERT-IN_DATA'
ORDER BY timestamp;

USERNAME   EXTENDED_TIMESTAMP                  OWNER      OBJ_NAME
ACTION_NAME
---------- ----------------------------------- ---------- ---------- ----
------------------------
AUDIT_CERT-IN_DATA 16-June-2007 14:16:55.435000 +00:00  AUDIT_CERT-
IN_DATA TEST_TAB   CREATE TABLE
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:16:55.514000 +00:00  AUDIT_CERT-
IN_DATA TEST_TAB   INSERT
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:16:55.545000 +00:00  AUDIT_CERT-
IN_DATA TEST_TAB   UPDATE
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:16:55.592000 +00:00  AUDIT_CERT-
IN_DATA TEST_TAB   SELECT
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:16:55.670000 +00:00  AUDIT_CERT-
IN_DATA TEST_TAB   DELETE
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:17:00.045000 +00:00  AUDIT_CERT-
IN_DATA TEST_TAB   DROP TABLE

6 rows selected.

SQL>
```

When the audit trail is directed to an XML format OS file, it can be read using a text editor or via the V$XML_AUDIT_TRAIL view, which contains similar information to the DBA_AUDIT_TRAIL view

```
COLUMN db_user       FORMAT A10
COLUMN object_schema FORMAT A10
COLUMN object_name   FORMAT A10
COLUMN extended_timestamp FORMAT A35

SELECT db_user,
       extended_timestamp,
       object_schema,
       object_name,
       action
FROM   v$xml_audit_trail
WHERE  object_schema = 'AUDIT_CERT-IN_DATA'
ORDER BY extended_timestamp;

DB_USER    EXTENDED_TIMESTAMP                  OBJECT_SCH
OBJECT_NAM     ACTION
---------- ----------------------------------- ---------- --------
-- ----------
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:14:33.417000 +00:00
AUDIT_CERT-IN_DATA TEST_TAB            1
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:14:33.464000 +00:00
AUDIT_CERT-IN_DATA TEST_TAB            2
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:14:33.511000 +00:00
AUDIT_CERT-IN_DATA TEST_TAB            6
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:14:33.542000 +00:00
AUDIT_CERT-IN_DATA TEST_TAB            3
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:14:33.605000 +00:00
AUDIT_CERT-IN_DATA TEST_TAB            7
AUDIT_CERT-IN_DATA 16-JUNE-2007 14:14:34.917000 +00:00
AUDIT_CERT-IN_DATA TEST_TAB           12

6 rows selected.

SQL>
```

Several fields were added to both the standard and fine-grained audit trails in Oracle 10g, including:

- EXTENDED_TIMESTAMP - A more precise value than the exising TIMESTAMP column.
- PROXY_SESSIONID - Proxy session serial number when an enterprise user is logging in via the proxy method.
- GLOBAL_UID - Global Universal Identifier for an enterprise user.
- INSTANCE_NUMBER - The INSTANCE_NUMBER value from the actioning instance.
- OS_PROCESS - Operating system process id for the oracle process.
- TRANSACTIONID - Transaction identifier for the audited transaction. This column can be used to join to the XID column on the FLASHBACK_TRANSACTION_QUERY view.
- SCN - System change number of the query. This column can be used in flashback queries.
- SQL_BIND - The values of any bind variables if any.
- SQL_TEXT - The SQL statement that initiated the audit action.

The SQL_BIND and SQL_TEXT columns are only populated when the AUDIT_TRAIL parameter is set to db,extended or xml,extended.

## 9.6 Maintenance and Security

Auditing should be planned carefully to control the quantity of audit information and be refined to match specific requirements; audit only specific operations or objects of interest.

The database audit trail must be deleted, or archived, on a regular basis to prevent the SYS.AUD$ table growing to an unnacceptable size.Only DBAs should have maintenance access to the audit trail. Auditing modifications of the data in the audit trail itself can be achieved using the following statement:

AUDIT INSERT, UPDATE, DELETE ON sys.aud$ BY ACCESS;

The OS and XML audit trails are managed through the OS. These files should be secured at the OS level by assigning the correct file permissions.

## 9.7 Fine Grained Auditing (FGA)

Fine grained auditing extends Oracle standard auditing capabilities by allowing the administrator to audit actions based on user-defined predicates. It is independent of the AUDIT_TRAIL parameter setting and all audit records are stored in the FGA_LOG$ table, rather than the AUD$ table. The following example illustrates how fine grained auditing is being used.

First, create a test table.   N m

```
CONN AUDIT_CERT-In_DATA/password
CREATE TABLE emp (
 ename     VARCHAR2(10),
 job       VARCHAR2(9),
 mgr       NUMBER(4),
 hiredate  DATE,
 sal       NUMBER(7,2),
 comm      NUMBER(7,2),
 deptno    NUMBER(2)
);

INSERT INTO emp (empno, ename, sal) VALUES (9999, 'Tim', 1);
INSERT INTO emp (empno, ename, sal) VALUES (9999, 'Larry',
50001);
COMMIT;
```

The following policy audits any queries of salaries greater than £50,000.

```
CONN sys/password AS sysdba

BEGIN
  DBMS_FGA.add_policy(
    object_schema   => 'AUDIT_CERT-IN_DATA',
    object_name     => 'EMP',
    policy_name     => 'SALARY_CHK_AUDIT',
    audit_condition => 'SAL > 50000',
    audit_column    => 'SAL');
END;
/
```

Querying both employees proves the auditing policy works as expected.

```
CONN AUDIT_CERT-In_DATA/password
SELECT sal FROM emp WHERE ename = 'Tim';
SELECT sal FROM emp WHERE ename = 'Larry';

CONN sys/password AS SYSDBA
SELECT sql_text
FROM   dba_fga_audit_trail;

SQL_TEXT
------------------------------------------
SELECT sal FROM emp WHERE ename = 'Larry'

1 row selected.

SQL>
```

Extra processing can be associated with an FGA event by defining a database procedure and associating this to the audit event. The following example assumes the FIRE_CLERK procedure has been defined:

```
BEGIN
  DBMS_FGA.add_policy(
    object_schema   => 'AUDIT_CERT-IN_DATA',
    object_name     => 'EMP',
    policy_name     => 'SALARY_CHK_AUDIT',
    audit_condition => 'SAL > 50000',
    audit_column    => 'SAL',
    handler_schema  => 'AUDIT_CERT-IN_DATA',
    handler_module  => 'FIRE_CLERK',
    enable          => TRUE);
END;
/
```

The DBMS_FGA package contains the following procedures:

- ADD_POLICY
- DROP_POLICY
- ENABLE_POLICY
- DISABLE_POLICY

In Oracle9i fine grained auditing has limited queries, but in Oracle 10g it has been extended to include DML statements, as shown by the following example.

- Clear down the audit trail.

```
CONN sys/password AS SYSDBA
TRUNCATE TABLE fga_log$;
SELECT sql_text FROM dba_fga_audit_trail;

no rows selected.
```

- Apply the policy to the SAL column of the EMP table.

```
BEGIN
  DBMS_FGA.add_policy(
    object_schema   => 'AUDIT_CERT-IN_DATA',
    object_name     => 'EMP',
    policy_name     => 'SAL_AUDIT',
    audit_condition => NULL, -- Equivalent to TRUE
    audit_column    => 'SAL',
    statement_types => 'SELECT,INSERT,UPDATE,DELETE');
END;
/
```

- **Test the auditing.**

```
CONN AUDIT_CERT-In_DATA/password
SELECT * FROM emp WHERE empno = 9998;
INSERT INTO emp (empno, ename, sal) VALUES (9998, 'Bill', 1);
UPDATE emp SET sal = 10 WHERE empno = 9998;
DELETE emp WHERE empno = 9998;
ROLLBACK;
```

- **Check the audit trail.**

```
CONN sys/password AS SYSDBA
SELECT sql_text FROM dba_fga_audit_trail;

SQL_TEXT
--------------------------------------
SELECT * FROM emp WHERE empno = 9998
INSERT INTO emp (empno, ename, sal) VALUES (9998, 'Bill', 1)
UPDATE emp SET sal = 10 WHERE empno = 9998
DELETE emp WHERE empno = 9998

4 rows selected.
```

- **Drop the policy.**

```
CONN sys/password AS SYSDBA
BEGIN
  DBMS_FGA.drop_policy(
    object_schema   => 'AUDIT_CERT-IN_DATA',
    object_name     => 'EMP',
    policy_name     => 'SAL_AUDIT');
END;
/
```

# 10. Auditing for Database Server - MS SQL Server 2005

## 10.1 SQL Server Auditing

Auditing helps to track unauthorized user behavior on systems. Auditing is especially useful to protect against rogue administrators or users with elevated privileges. A good auditing system allows filtering for only the events that deem important, so they are not inundated with information. All of the database platforms provide for auditing in varying degrees. SQL Server 2005 supports auditing in following ways:

- The SQL Profiler.
- Microsoft SQL Server Connection Auditing
- Auditing Windows Groups from SQL Server
- DDL Audit Mechanisms

## 10.2 The SQL Profiler Setup

SQL Profiler is a graphical tool that allows system administrators to monitor events in an instance of Microsoft SQL Server. It helps to capture and save data about each event to a file or SQL Server table to analyze later. For example, an administrator can monitor a production environment to see which stored procedures are hampering performance by executing too slowly.

Use SQL Profiler to monitor only the events which are of interest. If traces are becoming too large, administrator can filter them based on the information needed, so that only a subset of the event data is collected. Monitoring too many events adds overhead to the server and the monitoring process. Consequently, it can cause the trace file or trace table to grow very large, especially when the monitoring process takes place over a long period of time.

Administrator can configure the events by the process given below:

### 10.2.1 Start SQL Server 2005 Profiler

Start SQL Server 2005 Profiler via the GUI interface by navigating to Start | All Programs | Microsoft SQL Server 2005 | Performance Tools | SQL Server Profiler

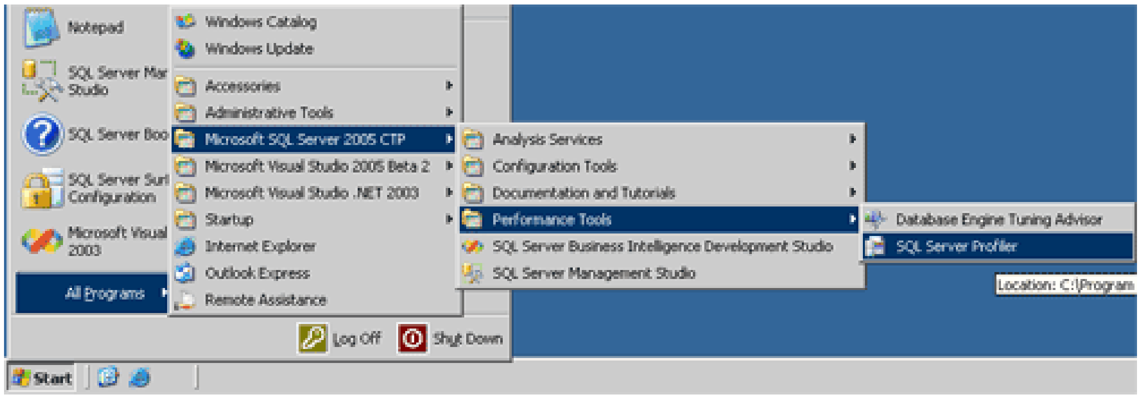
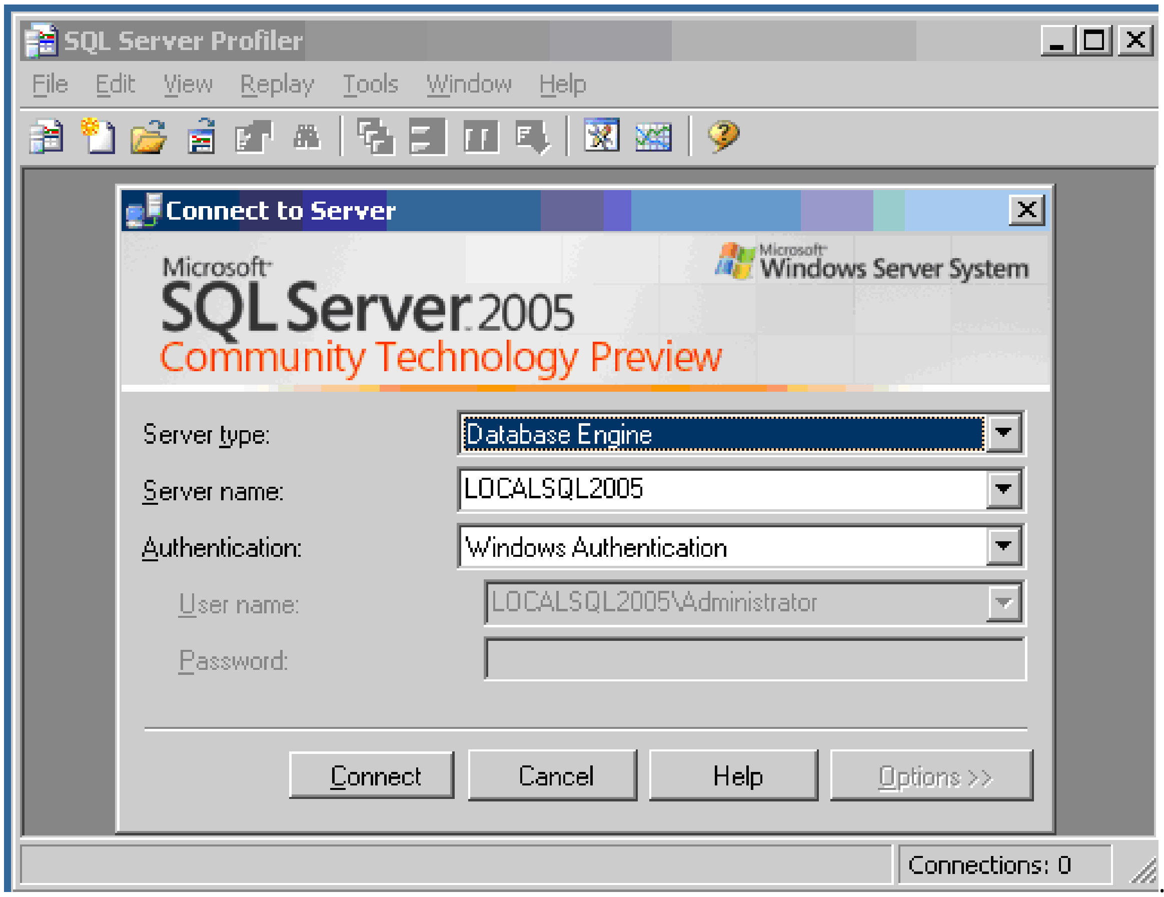

Notepad
SQL Server Management Studio
SQL Server Books
SQL Server Surface Area Configuration
Microsoft Visual 2003
All Programs
Windows Catalog
Windows Update
Accessories
Administrative Tools
Microsoft SQL Server 2005 CTP
Microsoft Visual Studio 2005 Beta 2
Microsoft Visual Studio .NET 2003
Startup
Internet Explorer
Outlook Express
Remote Assistance
Analysis Services
Configuration Tools
Documentation and Tutorials
Performance Tools
SQL Server Business Intelligence Development Studio
SQL Server Management Studio
Database Engine Tuning Advisor
SQL Server Profiler
Location: C:\Program
Log Off
Shut Down
Start

Figure-7: SQL Server 2005 Profiler

### 10.2.2 Start a new Trace session

Start a new Trace session by selecting the 'File' menu and the 'New Trace' option. Once the 'Connect to Server' interface loads, select the 'Type' as either 'Database Engine' or 'Analysis Services'. For example, use the ‘Database Engine’ option; select the 'Server Name' followed by the 'Authentication' type. Finally, click 'Connect' button to start the configuration.

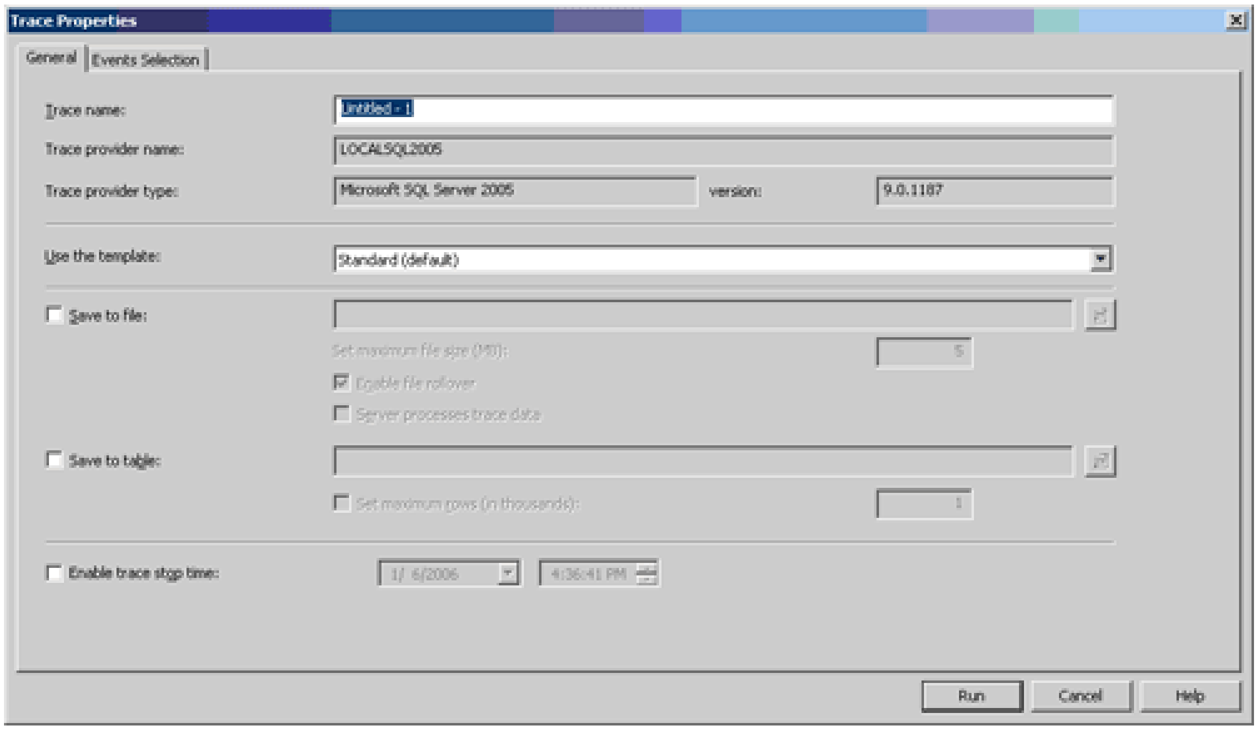


Server type: Database Engine
Server name: LOCALSQL2005
Authentication: Windows Authentication
User name: LOCALSQL2005\Administrator
Password:

Figure-8: Trace Session

### 10.2.3 Configure SQL Server Profiler Trace Properties – General Tab:

- Trace Name: Specify a meaningful name for the session
- Use the template: A total of eight templates are available with predefined events selected
- For example, use the 'Standard (default)' template
- Save to file or Save to table: To retain a copy of the data save the results to either a database table or Windows file.
- Save the results to the dbo.TraceResults table in a user defined database.

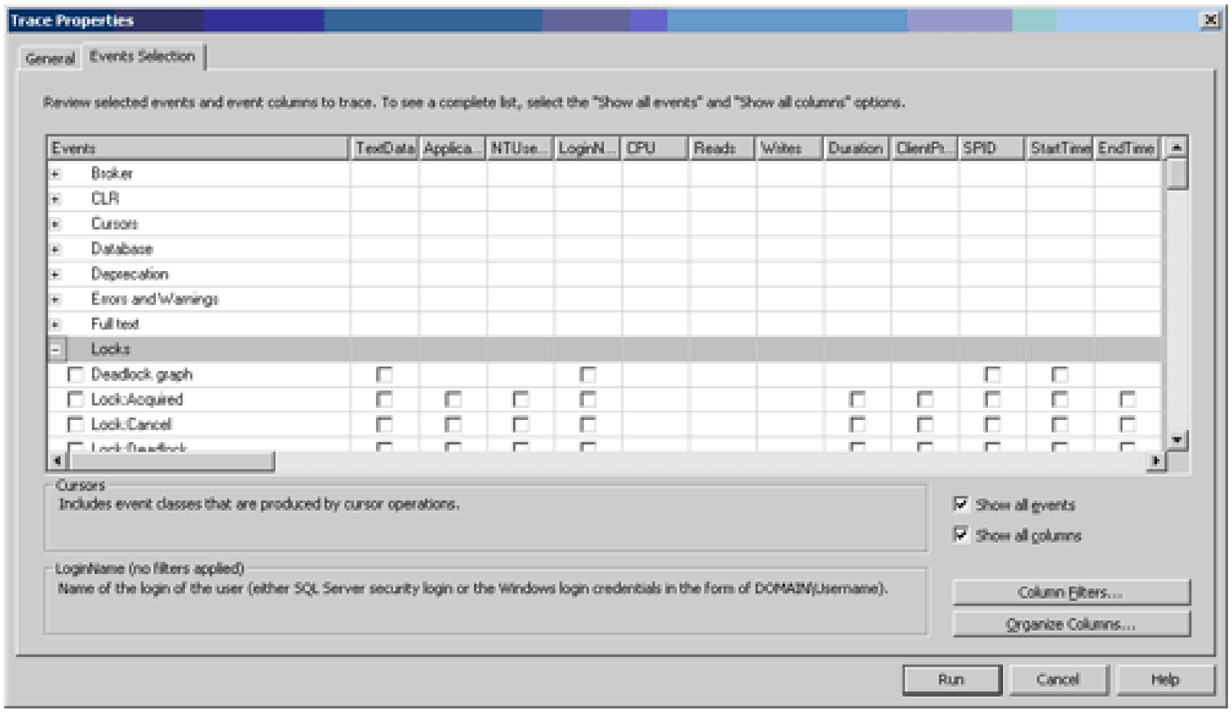


Trace name: Untitled - 1
Trace provider name: LOCALSQL2005
Trace provider type: Microsoft SQL Server 2005  version: 9.0.1187
Use the template: Standard (default)
☐ Save to file:
Set maximum file size (MB): 5
☑ Enable file rollover
☐ Server processes trace data
☐ Save to table:
☐ Set maximum rows (in thousands): 1
☐ Enable trace stop time: 1/ 6/2006  4:36:41 PM

Figure-9: Trace Properties - General

### 10.2.4 Configure SQL Server Profiler Trace Properties – Events Selection Tab

- Review the specific events and select the required check boxes to capture the desired data.
- Show All Events: Select this check box to see all events that Profiler will be able to capture.
- Show All Columns: Select this check box to see all columns that Profiler will be able to capture.



Trace Properties

General | Events Selection

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

| Events | TextData | Applica... | NTUse... | LoginN... | CPU | Reads | Writes | Duration | ClientPr... | SPID | StartTime | EndTime |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| + Broker | | | | | | | | | | | | |
| + CLR | | | | | | | | | | | | |
| + Cursors | | | | | | | | | | | | |
| + Database | | | | | | | | | | | | |
| + Deprecation | | | | | | | | | | | | |
| + Errors and Warnings | | | | | | | | | | | | |
| + Full text | | | | | | | | | | | | |
| − Locks | | | | | | | | | | | | |
| ☐ Deadlock graph | ☐ | | | ☐ | | | | | | ☐ | ☐ | |
| ☐ Lock:Acquired | ☐ | ☐ | ☐ | ☐ | | | | ☐ | ☐ | ☐ | ☐ | ☐ |
| ☐ Lock:Cancel | ☐ | ☐ | ☐ | ☐ | | | | ☐ | ☐ | ☐ | ☐ | ☐ |

Cursors
Includes event classes that are produced by cursor operations.

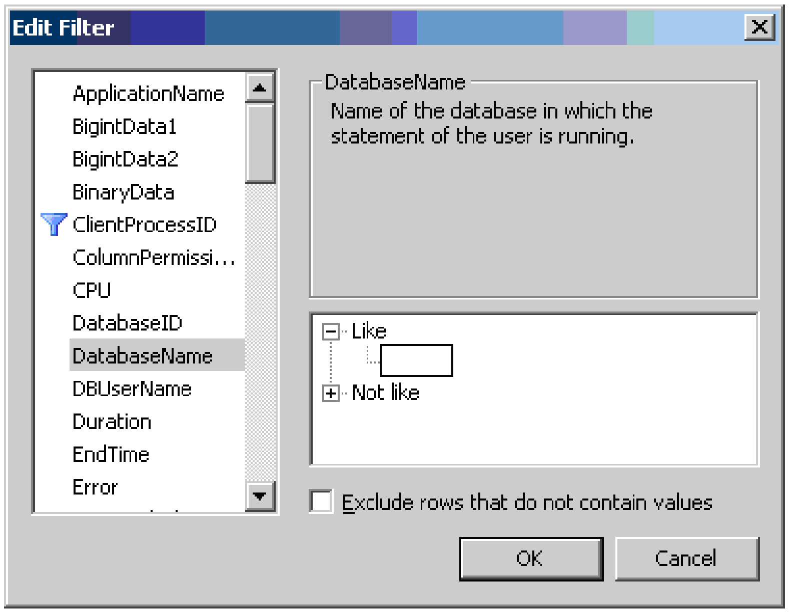
LoginName (no filters applied)
Name of the login of the user (either SQL Server security login or the Windows login credentials in the form of DOMAIN\Username).

☑ Show all events
☑ Show all columns

Figure-10: Trace Properties - Events Selection

### 10.2.5 Additional Configuration – Column Filters

- Specify filters based on the columns that are selected for the session to limit the data.



Edit Filter

ApplicationName
BigintData1
BigintData2
BinaryData
ClientProcessID
ColumnPermissi...
CPU
DatabaseID
DatabaseName
DBUserName
Duration
EndTime
Error

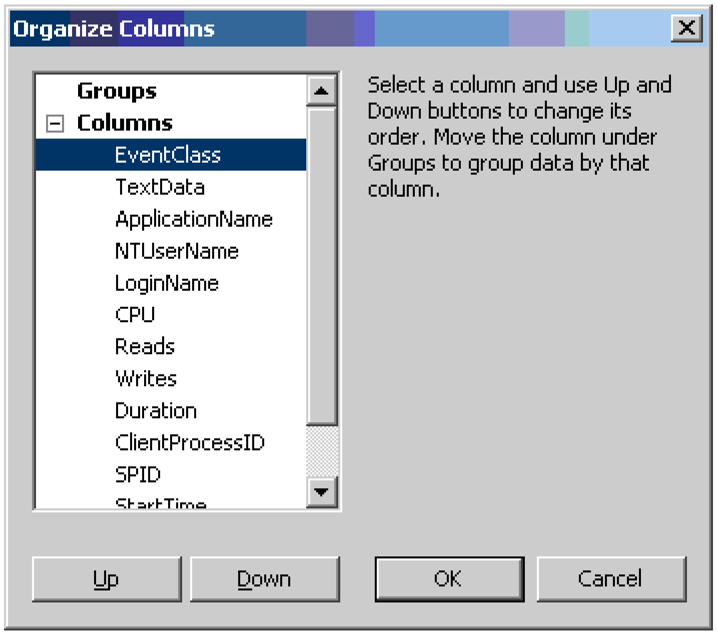
DatabaseName
Name of the database in which the statement of the user is running.

⊟ Like
⊞ Not like

☐ Exclude rows that do not contain values

Figure-11: Column Filter

- Additional Configuration – Organize Columns
  Specify the column order as well grouping settings for the final data.



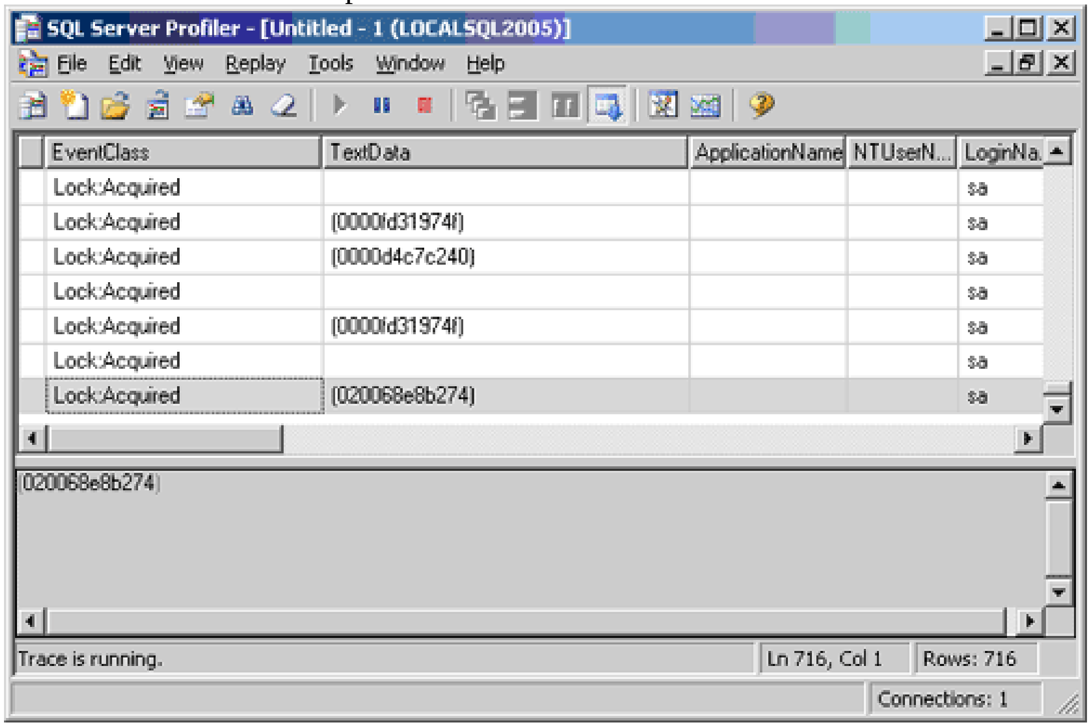
Organize Columns

Groups
Columns
EventClass
TextData
ApplicationName
NTUserName
LoginName
CPU
Reads
Writes
Duration
ClientProcessID
SPID
StartTime

Select a column and use Up and Down buttons to change its order. Move the column under Groups to group data by that column.

Figure-12: Organize Columns

- To start the session, click the 'Run' button on the Trace Properties interface (see Figure-10: Trace Properties - Events Selection)
- Review the results that are captured in the Profiler interface



SQL Server Profiler - [Untitled - 1 (LOCALSQL2005)]

| EventClass | TextData | ApplicationName | NTUserN... | LoginNa |
|---|---|---|---|---|
| Lock:Acquired | | | | sa |
| Lock:Acquired | (0000fd31974f) | | | sa |
| Lock:Acquired | (0000d4c7c240) | | | sa |
| Lock:Acquired | | | | sa |
| Lock:Acquired | (0000fd31974f) | | | sa |
| Lock:Acquired | | | | sa |
| Lock:Acquired | (020068e8b274) | | | sa |

(020068e8b274)

Trace is running. Ln 716, Col 1 Rows: 716

Connections: 1

Figure-13: SQL Server 2005 Profiler – Review Results

### 10.2.6 T-SQL queries to analyze the SQL Server Profiler results

**Total transactions per database in the last one hour**

```
SELECT COUNT(*) AS 'Total Records',
DB_NAME(DatabaseID) AS 'Database Name'
FROM dbo.TraceResults (NOLOCK)
WHERE StartTime BETWEEN DATEADD(Hour, -1,
GETDATE()) AND GETDATE()

GROUP BY DatabaseID
ORDER BY DatabaseID
GO
```

**High CPU usage in the last one hour**

```
USE CERT-In_Works
GO
SELECT * FROM dbo.TraceResults (NOLOCK)
WHERE CPU > 5000 AND StartTime
 BETWEEN DATEADD(Hour, -1, GETDATE()) AND GETDATE()
GO
```

**Long duration in the last one hour**

```
USE CERT-In_Works
GO
SELECT * FROM dbo.TraceResults (NOLOCK)
WHERE Duration > 10000000
AND StartTime BETWEEN DATEADD (Hour, -1, GETDATE ()) AND GETDATE ()
GO
```

### 10.2.7 Creating Traces

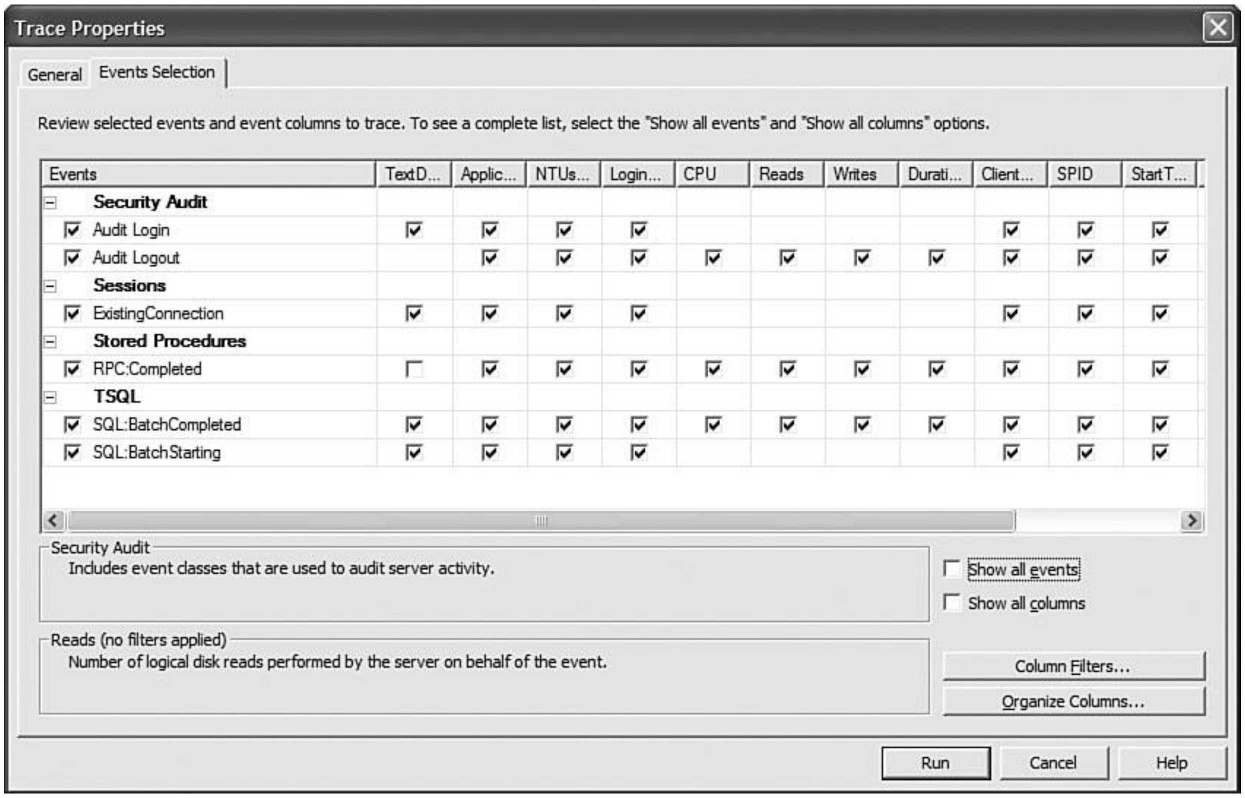
Because SQL Server Profiler can trace numerous events, it is easy to get lost when reading the trace output. It roughly determines the information and how the information has been grouped. For example, if the SQL statements that each user is submitting through an application, it could trace incoming SQL statements and group them by user and by application.

The Trace can be launched by selecting

- Start | SQL Server 2005 | Performance Tools | SQL Server Profiler. [It can also be launched from within SSMS (SQL Server Management Studio) from the Tools menu.]
- Refer steps given in section 10.2.2 to 10.2.4 for configuring a ‘New Trace’

### 10.2.8 Events

The events and data columns that will be captured by Profiler trace are defined on the Events Selection tab.



Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

| Events | TextD... | Applic... | NTUs... | Login... | CPU | Reads | Writes | Durati... | Client... | SPID | StartT... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ⊟ **Security Audit** | | | | | | | | | | | |
| ✓ Audit Login | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | ✓ | ✓ |
| ✓ Audit Logout | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ⊟ **Sessions** | | | | | | | | | | | |
| ✓ ExistingConnection | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | ✓ | ✓ |
| ⊟ **Stored Procedures** | | | | | | | | | | | |
| ✓ RPC:Completed | ☐ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ⊟ **TSQL** | | | | | | | | | | | |
| ✓ SQL:BatchCompleted | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ✓ SQL:BatchStarting | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | ✓ | ✓ |

Security Audit
Includes event classes that are used to audit server activity.

Reads (no filters applied)
Number of logical disk reads performed by the server on behalf of the event.

Figure-14:The Events Selection tab

The Events Selection tab has changed dramatically in SQL Server 2005. It consolidates the selection of events, data columns, and filters on one tab. In SQL Server 2000, there were three separate tabs for each of these elements. One of the biggest advantages of the SQL Server 2005 Events Selection tab is that the administrator can easily determine which data columns will be populated for each event by looking at the columns that have check boxes available for the event.

SQL Server 2005 provides a way to selectively import a trace file into a trace table. When importing a trace file into a trace table, administrator can filter the data before it goes into the table as well as combine multiple files into a single trace table. Once the data is in a trace table, administrator can load the trace table into Profiler or write own queries and reports against the trace table for more detailed analysis than is possible in Profiler. Microsoft SQL Server includes some built-in user-defined functions for working with Profiler traces. The fn_trace gettable function is used to import trace file data into a trace table. The following is the syntax for this function:

```
fn_trace_gettable( [ @filename = ] filename , [ @numfiles = ] number_files )
```

### 10.2.9 Creating and Inserting Trace Data into a Trace Table from a Trace File

```
SELECT  EventClass,
        EventSubClass,
        TextData = convert(varchar(8000), TextData),
        BinaryData,  ApplicationName, Duration, StartTime, EndTime,
        Reads, Writes, CPU, ObjectID, IndexID,  NestLevel
        INTO  TraceTable

FROM :: fn_trace_gettable('c:\temp\sampletrace_20060826_0232.trc', default)
where TextData is not null or EventClass
         in  (16, -- Attention
              25, -- Lock:Deadlock
              27, -- Lock:Timeout
              33, -- Exception
              58, -- Auto Update Stats
              59, -- Lock:Deadlock Chain
              79, -- Missing Column Statistics
              80, -- Missing Join Predicate
              92, -- Data File Auto Grow
              93, -- Log File Auto Grow
              94, -- Data File Auto Shrink
              95) -- Log File Auto Shrink

Insert into TraceTable

(EventClass, EventSubClass,TextData, BinaryData,ApplicationName, Duration, StartTime,
EndTime, Reads, Writes,CPU, ObjectID, IndexID, nestlevel)

SELECT EventClass, EventSubClass,


TextData = convert(varchar(7900), TextData), BinaryData,

ApplicationName, Duration, StartTime, EndTime, Reads, Writes,

CPU, ObjectID, IndexID, nestlevel

FROM ::fn_trace_gettable('c:\temp\sampletrace_20060826_0108.trc', -1)
where TextData is not null
      or EventClass in
      (16, -- Attention
      25, -- Lock:Deadlock
      27, -- Lock:Timeout
      33, -- Exception
```

```
        58, -- Auto Update Stats
        59, -- Lock:Deadlock Chain
        79, -- Missing Column Statistics
        80, -- Missing Join Predicate
        92, -- Data File Auto Grow
        93, -- Log File Auto Grow
        94, -- Data File Auto Shrink
        95) -- Log File Auto Shrink
go
```

Once the trace file is imported into a trace table, administrator can open the trace table in Profiler or run own queries against the trace table from a Query Editor window in SSMS.

For example, the following query returns the number of lock timeouts encountered for each table during the period the trace was running:

```
SELECT  object_name(ObjectId), count(*)
from TraceTable
where EventClass = 27 -- Lock:Timout Event
group by object_name(ObjectId)
go
```

## 10.3 SQL Server connection auditing

Microsoft SQL Server connection auditing is a subsystem that can be used to record failed and successful login attempts on the server. Recording connection attempts is useful in being able to discover:

- Who is attempting to connect to the database?
- When an attack has taken place?

This form of auditing has little negative side effects since the amount of activity being audited is minimal. The auditing of connection attempts typically does not result in a significant performance impact on the database, and rarely creates an excessive amount of data written to the log. Because of the importance of knowing what logins are connecting to the database and the minimal impact recording a login causes, it is rarely a bad decision to audit connection attempts by this method.

The possible settings for login auditing are:

- None - logs no auditing information
- Success - only successful logins are logged
- Failure - only failed logins are logged
- All - both successful and failed logins are logged

The auditing information is written to the Microsoft SQL Server error logs and to the Windows event log. To enable auditing of logins, perform the following actions:

- Open SSMS(SQL Server Management  Studio) and connect to the database
- Right-click on the instance and select “Properties” from the popup menu
- Open the “Security” tab
- Under “Audit Level” choose “All”
- Click OK

**The Security Audit event category has the event classes described in the following table**

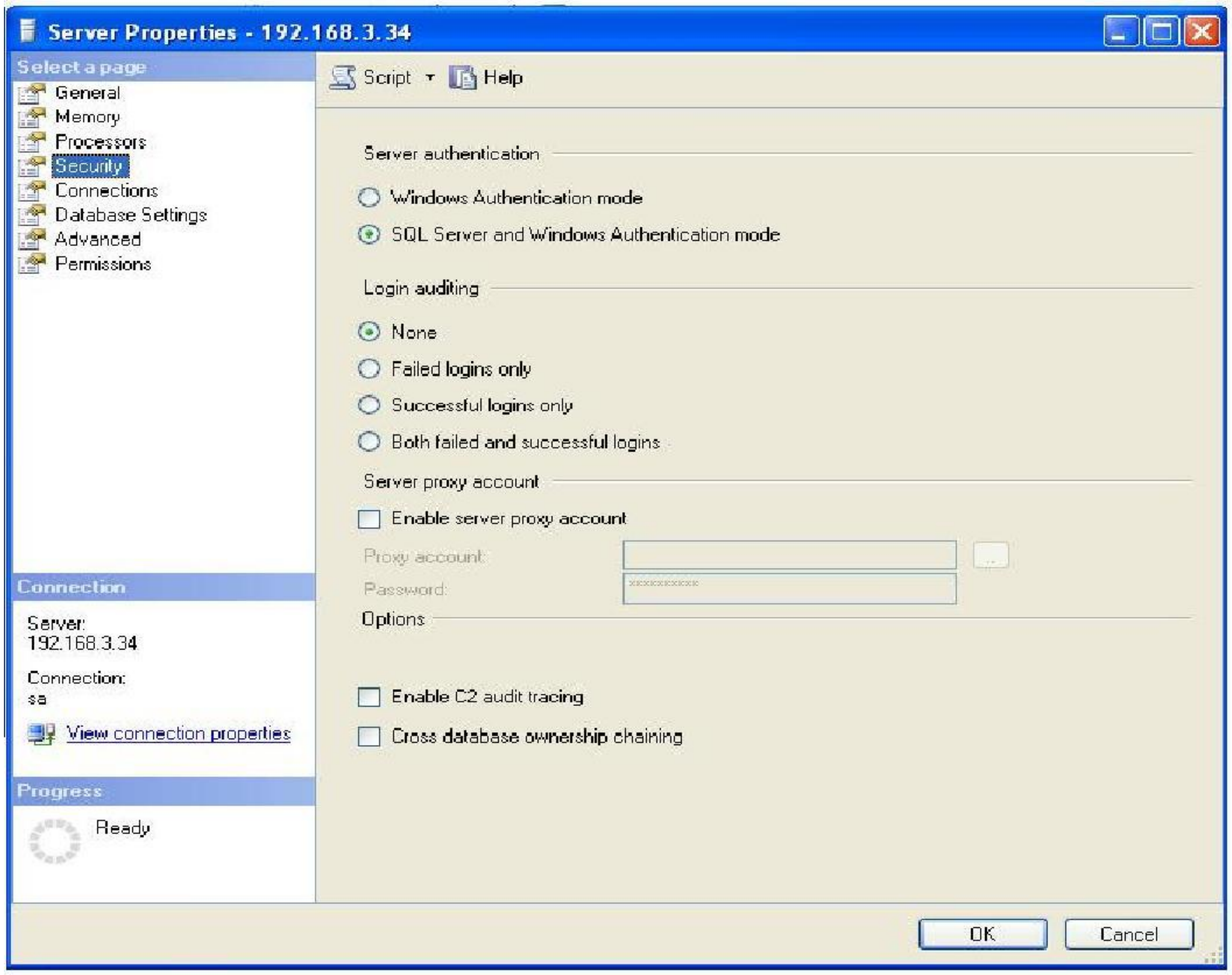
| **Event Class** | **Description** |
|---|---|
| Audit Login | Records all new connections events since the trace started such as when a client requested a connection to a server running on an instance of Microsoft Server 2005 Analysis Services (SSAS) |
| Audit Backup/Restore | Records server backup and restore events since the trace started such as Restore Events |
| Audit Logout | Records all new disconnect events since the trace started such as when a client issue a disconnect command Audit Server Start and Stop Records shut down ,start and pause activities for services.(There should be  a break in this sentence but I am unable to correct this) |
| Audit Object | Records all object permission changes |

Table-11: Event Class Description

**Use login auditing to monitor SQL Server Database Engine login activity**.

Login auditing can be configured to write to the error log on the following events.
- Failed logins
- Successful logins
- Both failed and successful logins

**To configure login auditing in SSMO (SQL Server Management Objects)**
- In SQL Server Management Studio, connect to an instance of the SQL Server Database Engine with Object Explorer.
- In Object Explorer, right-click the server name, and then click Properties.
- On the Security page, under Login auditing, click the desired option.



Server: 192.168.3.34

Connection: sa

Server authentication

Windows Authentication mode [✓ SQL Server and Windows Authentication mode]

Login auditing

[✓ None] Failed logins only Successful logins only Both failed and successful logins

Server proxy account

☐ Enable server proxy account

Proxy account:

Password: xxxxxxxxxx

Options

☐ Enable C2 audit tracing

☐ Cross database ownership chaining

Figure-15: SQL Server Management Objects (SSMO)

## C2 Audit Mode

C2 Audit Mode data is saved in a file in the \MSSQL\Data directory of default instances, and the \MSSQL$instancename\Data directory of named instances. If the audit log file reaches its size limit of 200 megabytes (MB), SQL Server will close the old file, create a new file, and write all new audit records to the new file. This process will continue until the audit data directory fills up or auditing is turned off.

C2 Audit Mode can be configured through SQL Server Management Studio or with sp_configure. Selecting this option will configure the server to record both failed and successful attempts to access statements and objects. This information can help in tracking user’s system activity (profile-based) and tracking possible security policy violations.

## 10.4 Auditing Windows Groups from SQL Server

SQL Server offers some insight into this issue with the xp_logininfo extended stored procedure. This stored procedure is part of both SQL Server 2000 and SQL Server 2005. This extended stored procedure takes the following parameters:

- @acctname - the windows account name or group
- @option - information to display
- 'all' - display information for all permission paths
    - 'members' - display list of members in a group
- @privilege - this is an output variable from this command and returns 'admin', 'user' or 'null

```
DECLARE @LoginName sysname
DECLARE @sql NVARCHAR (2000)

BEGIN
   DECLARE cur_Loginfetch CURSOR FOR

   SELECT [name] FROM master.sys.server_principals WHERE TYPE = 'G'

   OPEN cur_Loginfetch

   FETCH NEXT FROM cur_Loginfetch INTO @LoginName
   WHILE @@FETCH_STATUS = 0
       BEGIN
           EXEC xp_logininfo @LoginName , 'members'
           FETCH NEXT FROM cur_Loginfetch INTO @LoginName
       END
   CLOSE cur_Loginfetch
   DEALLOCATE cur_Loginfetch
   RETURN
```

### 10.4.1 Built-in\Administrators Group

- The principal of least privilege is used while granting users, developers, DBAs, network administrators, etc. the needed rights. If additional rights are required, evaluate the rights and grant the access accordingly.
- The BUILTIN\Administrators can easily be removed from SQL Server to prevent this security issue, but heed the warnings below prior to removing the group from SQL Server.

**Steps should take prior to considering removing this group**

- Verify that the "sa" password by logging into the SQL Server with the "sa" account with either Query Analyzer or Management Studio on the SQL Server

- Validate other groups or user accounts that are assigned SQL Server System with Administrator rights on the SQL Server.
- Review the rights assigned to the BUILTIN\Administrators group.
- Validate the members of the Windows Local Administrators group.
- Validate delegated Active Directory Group or user account with assigned rights in SQL Server and should be created only if necessary
- Validate that the members of the BUILTIN\Administrators group do not own any databases, objects, Jobs, etc and that none of the logins are connected to SQL Server.

| **Method** | **Directions** |
|---|---|
| SQL Server 2005<br>T-SQL Commands | DROP LOGIN [BUILTIN\Administrators] |
| SQL Server 2005<br>Management Studio | 1. Open Management Studio<br>2. Navigate to the Security folder<br>3. Expand (+) the Logins folder, which will load the SQL Server logins and groups in the right pane.<br>4. Locate the BUILTIN\Administrators group.<br>5. Right click on the BUILTIN\Administrators group and select the 'Delete' option.<br>6. On the subsequent screen, review the screen and if agree press 'OK' to remove the login.<br>7. Refresh the pane to verify the login has been dropped |

Table-12: Methods for removing Built-In\Administrators Group

**When was the last time the 'sa' password changed?**

Find out when the 'sa' password was last changed. If this timeframe is unacceptable to organization, then steps need to be taken to understand where the sa login is used and how the application can be modified to use another login.

```
USE Master
GO
SELECT [name], sid,
create_date, modify_date
FROM sys.sql_logins
WHERE [name] = 'sa'
GO
```

## 10.5 DDL (Data Definition Language) Audit Mechanisms

Data Definition Language (DDL) describes the portion of SQL that allows creating, altering, and destroying database objects. These database objects include schemas, tables, views, sequences, catalogs, indexes, and aliases.

### 10.5.1 Default Trace

Default trace provides troubleshooting assistance to database administrators by ensuring that they have the log data necessary to diagnose problems the first time they occur.

- By default controlled by sp_configure ‘default trace enabled’
- Stored in the same folder as the SQL Server Error Log
- Location can be changed by modifying the start up parameter using the SQL Server Configuration Manager
- Captures mainly audit type trace events
- The system function **fn_trace_gettable** has been enhanced to allow the reading of running server side traces
- SMO Trace API Microsoft.SqlServer.Management.Trace

### 10.5.2 DDL Audit Mechanisms – DDL Triggers

DDL triggers are a special kind of trigger that fire in response to Data Definition Language (DDL) statements. They can be used to perform auditing and regulating database operations.

- React to CREATE,ALTER,DROP DDL statements
- Server or Database scoped
- Within a DDL trigger user can rollback the DDL that caused it to fire
- Use the eventdata() function to return data as xml
- Synchronous

### 10.5.3 DDL Audit Mechanisms – Event Notifications

System stored procedures that perform DDL-like operations also fire DDL triggers and event notifications. It is recommended to test DDL triggers and event notifications to determine their responses to system stored procedures that are run.

- Leverage Service Broker infrastructure
- Asynchronous
- Target local or remote service
- React to DDL and subset of Trace events
- Integrate with WMI and SQL Agent alerts

### 10.5.4 Service Broker

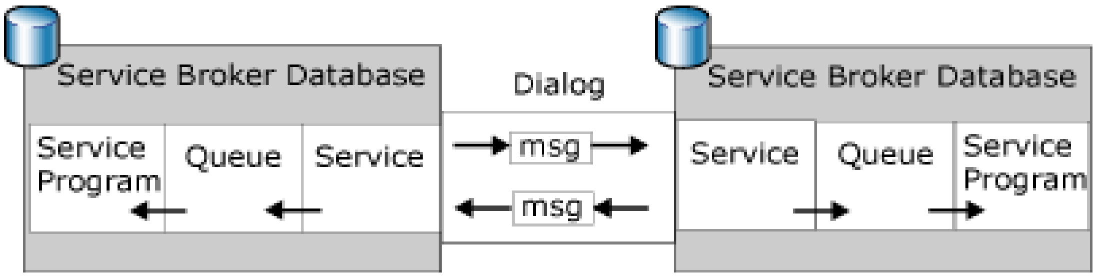


Figure-16: Service Broker

- Service Broker provides queuing and reliable messaging as part of the Database Engine
- Event Notification Service is built in to all databases

### 10.5.5 Event Notification Syntax

```
CREATE EVENT NOTIFICATION name
ON { SERVER | DATABASE | QUEUE }
[ WITH FAN_IN ]
FOR { event_type | event_group } [ ,...n ]
TO SERVICE broker_service
{ 'broker_instance_specifier' | 'current database' }
```

### 10.5.6 Event Groups

```
DDL_DATABASE_LEVEL_EVENTS
 └────────── DDL_TABLE_VIEW_EVENTS
   ├───────────── DDL_TABLE_EVENTS
   │          (CREATE_TABLE,ALTER_TABLE,DROP_TABLE)
   │
   ├───────────── DDL_VIEW_EVENTS
   │              (CREATE_VIEW,ALTER_VIEW,DROP_VIEW)
   │
   ├───────────── DDL_INDEX_EVENTS
   │              (CREATE_INDEX, ALTER_INDEX, DROP_INDEX)
   ├───────────── DDL_STATISTICS_EVENTS
   │ (CREATE_STATISTICS, ALTER_STATISTICS, DROP_STATISTICS)
```

### 10.5.7 Trace Events

In addition to the DDL events available in DDL Triggers, Event Notifications also allow a subset of **Trace events** to be captured

- Audit_Login
- Audit_Login_Failed
- Lock_Deadlock
- Data_File_Auto_Grow
- Blocked_Process_Report

### 10.5.8 Creating Event Notifications

- Create a **QUEUE**
- Create a **SERVICE** on a **QUEUE**
- Create a **ROUTE** for the **SERVICE**
- Create an **EVENT NOTIFICATION** to a **SERVICE**
- Create a **SERVICE PROGRAM** to process notification events in the **QUEUE**

### 10.5.9 Remote Event Notifications – Endpoint

- In order for Service Broker messages to pass between instances we need to create a Service Broker Endpoint
- Define the TCP port and encryption supported by the endpoint and entities that can be connected

```
CREATE ENDPOINT SSBENDPOINT
STATE = STARTED,
AS TCP (LISTENER PORT = 4022)
FOR SERVICE_BROKER
(AUTHENTICATION = WINDOWS,
 ENCRYPTION = SUPPORTED)
```

### 10.5.10 Remote Event Notifications – Routes

- Need to specify
    - Remote Service Name
    - Remote Broker Instance
    - Remote Address

```
CREATE ROUTE ExpenseRoute WITH
SERVICE_NAME = '//Adventure-Works.com/Expenses',
BROKER_INSTANCE = 'D8D4D268-00A3-4C62-8F91-634B89C1E315',
ADDRESS = 'TCP://SERVER02:1234'
```

- Need both an outbound route and a return route

### 10.5.11 Remote Event Notifications – Security

- Can configure full dialog security as described in BOL however it is overkill for DDL Audit and hard to automate (requires certificate management)
- Use Windows Authentication based on the SQL Server Service account with no encryption
- Grant connect on the Service Broker endpoints to the SQL Service accounts (local and remote) to allow communication
- NOTE guest user must have access to msdb (all object permissions can be denied)

### 10.5.12 Remote Event Notifications – DMV

- sys.transmission_queue in msdb holds outbound messages from audited instance.
- sys.conversation_endpoints describes service broker conversations and their states
- sys.server_event_notifications and sys.server_events hold metadata about event notifications
- sys.services and sys.routes hold service broker metadata

With SQL Server 2005 DDL (Data Definition Language) triggers have been introduced. They are different than INSERT, UPDATE and DELETE triggers, these triggers are fired when a change is made using commands such as ALTER, CREATE or DROP. The trigger is fired after the event occurs.

Creating a DDL trigger is equivalent to create a DML trigger. For example, following is a trigger that would fire whenever there is a DROP_TABLE or ALTER_TABLE event.

```
CREATE TRIGGER tr_tableChange
ON DATABASE
FOR DROP_TABLE, ALTER_TABLE
AS
BEGIN
   do some activity based on event
END
```

In addition to individual events, DDL events can be collected by using event groups.

This example here is triggered whenever there is DDL_LOGIN_EVENTS action. This includes DROP_LOGIN, CREATE_LOGIN and ALTER_LOGIN.

```
CREATE TRIGGER tr_LoginEvents
ON ALL SERVER
FOR DDL_LOGIN_EVENTS
AS
BEGIN
```

```
   do some activity based on event
END
```

## 10.10. Conclusion

SQL Server 2005 includes a role that allows users who do not have system administrator rights to use SQL Profiler. This will help developers and auditors to monitor the system without giving them extra rights. Windows Security Event log and SQL Server Profiler can cover most of the auditing needs, except for Data Definition Language (DDL) usage. DDL is the use of the SQL languages CREATE, ALTER, and DROP statements. SQL Server 2005 provides DDL triggers so auditor can determine when objects are created and deleted in the database and notifications through greater integration with the SQL Server Notification Services. SQL Profiler also includes new auditing capabilities for Microsoft SQL Server Analysis Services to improve auditing of OLAP data.

# 11. References and Resources

## General References and Resources

1. System Security guidelines
   http://www.cert-in.org.in/knowledgebase/guidelines/cisg-2004-04.htm
2. Web server Security guidelines
3. http://www.cert-in.org.in/knowledgebase/guidelines/cisg-2004-04.htm
4. NIST SP – 800-92 Guideline for Log Management
   www.csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf
5. http://www.loganalysis.org
6. http://www.syslog.org/
7. http://www.cert-in.org.in/knowledgebase/presentation/Logs-Forensics.pdf
8. http://www.cert-in.org.in/knowledgebase/presentation/16thapril04/loganalysis.pdf

## Windows Auditing and Logging

9. Threats and Countermeasures Guide
   http://www.microsoft.com/technet/security/guidance/serversecurity/tcg/tcgch00.mspx
10. http://technet2.microsoft.com/windowsserver/en/library/6847e72b-9c47-42ab-b3e3-691addac9f331033.mspx?mfr=true
11. http://technet2.microsoft.com/windowsserver/en/library/d8fc798c-1e77-4043-b59c-971b4961d85a1033.mspx?mfr=true
12. http://technet2.microsoft.com/windowsserver/en/library/42c66475-3346-428f-8faf-47a6611655ee1033.mspx?mfr=true
13. http://technet2.microsoft.com/windowsserver/en/library/20068d03-6473-4e00-84d4-fb1c7cce57d21033.mspx?mfr=true
14. http://technet2.microsoft.com/windowsserver/en/library/e104c96f-e243-41c5-aaea-d046555a079d1033.mspx?mfr=true
15. http://technet2.microsoft.com/windowsserver/en/library/50fdb7bc-7dae-4dcd-8591-382aeff2ea791033.mspx?mfr=true
16. http://technet2.microsoft.com/windowsserver/en/library/962f5863-15df-4271-9ae0-4b0412e297491033.mspx?mfr=true
17. http://technet2.microsoft.com/windowsserver/en/library/ee2f85ac-e3fb-4a24-b133-8c7fdfc5cee81033.mspx?mfr=true
18. http://technet2.microsoft.com/windowsserver/en/library/0a642c0c-387a-44f5-bfd9-951b87fd13801033.mspx?mfr=true
19. http://technet2.microsoft.com/windowsserver/en/library/a8297bc2-d53a-4a2f-94c5-8e412ae4e3861033.mspx?mfr=true
20. http://www.winsyslog.com/en/
21. http://www.eventid.net
22. http://www.monilog.com/en/
23. http://www.eventreporter.com/en

**Event Viewer**

24. http://technet2.microsoft.com/windowsserver/en/library/905f2810-a482-4040-8f28-1280d5ddd2d51033.mspx?mfr=true
25. http://technet2.microsoft.com/windowsserver/en/library/ad42d67d-f7f6-4aef-a130-dd4ea0dc93491033.mspx?mfr=true
26. http://technet2.microsoft.com/windowsserver/en/library/6fcff4d4-5624-43fc-aed1-68808d17d0391033.mspx?mfr=true

**Windows Event Logs**

27. http://support.microsoft.com/kb/308427
28. http://technet2.microsoft.com/windowsserver/en/library/9930c8f1-54ed-4d07-afa6-bc3c597bbe9c1033.mspx?mfr=true

**Configuring Logging Parameters in Windows**

29. http://technet2.microsoft.com/windowsserver/en/library/bb9fa858-a5bb-4769-93f3-65f2fbc04e3c1033.mspx?mfr=true
30. http://technet2.microsoft.com/windowsserver/en/library/62e90135-7a56-417b-a983-8deeff08a9a91033.mspx?mfr=true
31. http://technet2.microsoft.com/windowsserver/en/library/29e9aa80-8b6f-464c-9bbf-7f50b7034df81033.mspx?mfr=true
32. http://technet2.microsoft.com/windowsserver/en/library/63cf1837-2172-4996-a94d-db62023194151033.mspx?mfr=true
33. http://technet2.microsoft.com/windowsserver/en/library/1d920fd7-aac5-42a5-b92d-28ee7a30bc911033.mspx?mfr=true
34. http://technet2.microsoft.com/windowsserver/en/library/07b597d2-01cc-4663-8a98-c805948f399f1033.mspx?mfr=true
35. http://technet2.microsoft.com/windowsserver/en/library/d218436b-03d8-4bba-ac80-7ed5eec0d6c91033.mspx?mfr=true

**Retaining Event Logs**

36. http://technet2.microsoft.com/windowsserver/en/library/66f2f79c-9248-449a-811c-0a34ad6394751033.mspx?mfr=true
37. http://technet2.microsoft.com/windowsserver/en/library/1903d159-c1ee-486e-932f-cb8d881b94951033.mspx?mfr=true
38. http://technet2.microsoft.com/windowsserver/en/library/b837f6b4-9252-40f1-8952-f5b7fc2d90ab1033.mspx?mfr=true
39. http://technet2.microsoft.com/windowsserver/en/library/56d7a732-5730-4452-a701-270d4db56d961033.mspx?mfr=true
40. http://technet2.microsoft.com/windowsserver/en/library/a4b84910-7f82-4d50-b537-cc58a3e2acbc1033.mspx?mfr=true

41. http://technet2.microsoft.com/windowsserver/en/library/fb78928d-0329-43df-9700-383b7e53c8f11033.mspx?mfr=true

**IIS webserver Auditing and Logging**

42. Securing IIS 6.0 Web server http://www.cert-in.org.in/knowledgebase/guidelines/cisg-2006-01.htm
43. Microsoft Windows Server 2003 Deployment Kit - Deploying Internet Information Services (IIS) 6.0
44. Microsoft IIS 6.0: Administrator's Pocket Consultant - By William R. Stanek (Publisher: Microsoft Press)
45. Microsoft® IIS 6 Delta Guide - By Martin C. Brown (Publisher : Sams Publishing)
46. IIS 6 Administration - By Mitch Tulloch (Publisher: Osborne/McGraw-Hill)
47. Microsoft Windows Server 2003 Deployment Kit: Designing and Deploying Directory and Security Services by Microsoft Corporation (Publisher: Microsoft Press)
48. Securing IIS 6.0 - By Chun Hai (Bernard) Cheah Ken Schaefer Chris Peiris

**Linux system and Apache webserver Auditing and Logging**

49. http://www.redhat.com/docs/manuals/cert-system/admin/7.2/index.html
50. Implementation of Central Logging Server using syslog-ng http://www.cert-in.org.in/knowledgebase/guidelines/cisg-2004-03.htm
51. http://httpd.apache.org/
52. http://httpd.apache.org/docs/2.2/logs.html
53. Introduction to the Apache Web Server, Rich Bowen, Asbury College
54. Secure Configuration of the Apache Web Server, MITRE, Center for Integrated Intelligence Systems

**MS SQL Auditing**

55. http://www.microsoft.com/technet/prodtechnol/sql/themes/security.mspx
56. Unleashed SQL Server 2005 - SAMS Publishing
57. http://www.microsoft.com/technet/prodtechnol/sql/2005/sql2005secbestpract.mspx
58. http://blog.sqlauthority.com/2007/03/21/sql-server-2005-security-best-practices-operational-and-administrative-tasks/
59. http://www.sql-server-performance.com/articles/audit/ddl_triggers_p3.aspx
60. Beginning-sql-server-2005-administration - WROX Publishing
61. http://www.sqldbatips.com/presentations/PASS2006.ppt#1
62. http://msdn2.microsoft.com/en-us/ms189540.aspx
63. http://www.appsecinc.com/presentations/Security_Auditing_MSSQL.pdf

**Oracle Auditing**

64. http://www.oracle.com/
65. http://www.metalink.oracle.com/
66. http://www.petefinnigan.com/orasec.htm
67. Introduction to Simple Oracle Auditing, Pete Finnigan
    http://www.securityfocus.com/infocus/1689

**Common Syslog Sever Implementations**

68. syslog-ng
    http://www.balabit.com/network-security/syslog-ng
69. Kiwi Syslog
    http://www.kiwisyslog.com/kiwi-syslog-daemon-overview/
70. Metalog
    http://metalog.sourceforge.net/
71. Modular Syslog (Msyslog)
    http://sourceforge.net/projects/msyslog/
72. nsyslog
    http://coombs.anu.edu.au/~avalon/nsyslog.html
73. Rsyslog
    http://www.rsyslog.com/
74. San Diego Supercomputer Center (SDSC) Secure Syslog
    http://sourceforge.net/projects/sdscsyslog/
75. Winsyslog
    http://www.winsyslog.com/en/

**Common Security Information and Event Management (SIEM) Products**

76. ArcSight SIEM Platform
    http://www.arcsight.com/product_overview.htm
77. SecureVue: Enterprise Security Management
    http://www.eiqnetworks.com/products/SecureVue/Enterprise_Security_Management.shtml
78. RSA enVision platform
    http://www.rsa.com/node.aspx?id=3170
79. Cisco Security Monitoring, Analysis, and Response System (MARS)
    http://www.cisco.com/en/US/products/ps6241/index.html
80. Symantec Security Information Manager
    http://www.symantec.com/business/security-information-manager
81. Symantec Control Compliance Suite
    http://www.symantec.com/business/control-compliance-suite

82. CA Audit
    http://www.ca.com/us/products/product.aspx?id=157
83. CA Security Command Center
    http://www.ca.com/us/products/product.aspx?id=4351
84. NetIQ Security Manager
    http://www.netiq.com/products/sm/default.asp?id=Overview
85. Unified Security Monitoring
    http://www.tenablesecurity.com/products/
86. Log Parser
    http://www.microsoft.com/technet/scriptcenter/tools/logparser/default.mspx
87. Logwatch
    http://www.logwatch.org/
88. TriGeo Security Information Manager
    http://www.trigeo.com/products/
89. SNARE
    http://www.intersectalliance.com/
90. QRadar Simple Log Management Information Manager (SLIM)
    http://www.q1labs.com/products/408/qradar-slim/
91. EventTracker
    http://www.prismmicrosys.com/EventTracker/

---

Feedback: CERT-In welcomes feedback on this guideline. Feedback may be sent to info@cert-in.org.in.