## Centralized Audit Framework

This project was a collaboration between Caregroup Healthcare (Ayad Shammout), Microsoft Consulting Services (Andy Roberts), and SQL Customer Advisory Team (Denny Lee). The goal was to create a working template of an end-to-end audit project that obtains the audit logs, transforms the data, loads the data, aggregates the data, and build your reports. For more information, see the (to be published) *Audit Project Technical Spotlight*.

**Figure 1:** Auditing Project Solution workflow



**Acquire the Solution**

Unzip SQLAudit.zip. It should contain the following files:

* SQLAuditRepositoryDatabase.sql – the SQL file that creates your SQLAudit database
* LoadLogsPackage.dtsx – an SSIS package that performs the task of obtaining, extracting, transforming, and loading the audit data and archiving the audit logs
* SQLAuditReports– A SQL Server Reporting Services (SSRS) solution that you can deploy onto your server so you can view the audit reports

**Create the SQLAudit Database**

This database will be the repository of your audit data.

1. Open the file SQLAuditRepositoryDatabase.sql in SQL Server Management Studio and enable SQLCMD mode (on the **Query** menu, select SQLCMD mode). You will know you’re in SQLCMD mode if your :setvar lines are highlighted in grey.
2. Define the following variables.

|  |  |  |
| --- | --- | --- |
| Variable | Description | Comments |
| DataDirectory | SQL DB data folder (e.g., H:\sqldata\) | Make sure the path ends with a slash “\”. |
| LogDirectory | SQL DB log folder (e.g., H:\sqllog\) | Make sure the path ends with a slash “\”. |
| DatabaseName | Name of your database (e.g., SQLAudit) | None. |

1. Execute the script; your database is created.

**Install the LoadLogsPackageSSIS package**

You will create the configuration files necessary for your SSIS package so that you can execute it.

1. Double click **SQLAuditLoader.SSISDeploymentManifest**. If it doesn't have an associated program, navigate to C:\program files\Microsoft SQL Server\100\DTS\Binn\dtsinstall.exe to open it in the Package Installation Wizard. For more information about installing packages, see <http://msdn.microsoft.com/en-us/library/ms365321(SQL.100).aspx>
2. Click **Next**, click **File system Deployment**, and then click **Next**. Choose an installation folder such as C:\Program Files\Microsoft SQL Server\100\DTS\Packages\SQLAuditLoader, click **Next**, and then click **Next**.
3. Configure the following package properties.

|  |  |
| --- | --- |
| Parameter | Example value |
| **SSIS log provider for SQL Server** | SqlAuditLogRepository |
| **SqlAuditLogRepository**  Provide the name of your SQLAudit database (e.g., SQLAudit) | Data Source=.;Initial Catalog=**$DBName$**;Provider=SQLNCLI10.1;Integrated Security=SSPI;Auto Translate=False;Application Name=SSIS-Package-{21C9032A-E45A-41F2-BA67-9EF35FCD18C3}SqlAuditLogRepository; |
| **User:auditLogArchivePath**  Location to archive the audit logs after processing | D:\audit\logs\archive |
| **User:LogFilePath**  Location of your audit logs | D:\Audit\logs |

These values will be stored in the **LoadLogsPackageConfig.dtsConfig** file. You can alter them manually if you need to make changes after you install the SSIS package.

**Note:** This version of the Audit project assumes a local file directory to obtain the audit logs.

1. Click **Next**, and then click **Finish**.

**Execute the SSIS Package**

When the package is executed, it assumes that there are log files in the D:\audit\logs location (as noted above) and after the log files have been processed, they will be moved to the D:\audit\logs\archive folder.

1. From a command prompt, change to your SQLAuditLoader directory (e.g., C:\Program Files\Microsoft SQL Server\100\DTS\Packages\SQLAuditLoader)
2. Execute the command:

dtexec /ConfigFile LoadLogsPackageConfig.dtsConfig /File LoadLogsPackage.dtsx

**Note:** Make sure you are executing SQL Server 2008 version of DTExec. On servers with multiple instances of SQL Server 2005 and SQL Server 2008, if you run DTExec, DTExect may default to the SQL Server 2005 version (9.00.xxxx) instead of the SQL Server 2008 version (10.00.xxxx). To avoid this, you can fully qualify the path of DTExec (e.g., C:\Program Files\Microsoft SQL Server\100\DTS\Binn\dtexec.exe).

Because there are a lot of audit log files, it is suggested that you run this SSIS package periodically (e.g., every 15 minutes) to ensure that the data is loaded in a timely manner.

**Generate Your Aggregate Reports**

Executing the SSIS package will take the audit logs and populate the aud.AuditLog\_[EventType] tables within your SQLAudit database. However, because there are a lot of audit events, it is beneficial to create aggregate reports (i.e., summary reports) that provide a summary view of all of your audits.

* To do this, in SQL Server Management Studio, connect to your SQLAudit database, and then execute the following commands.

exec aud.rspAggServerActions @EventDate = '08/22/2008'

exec aud.rspAggDatabaseActions @EventDate = '08/22/2008'

exec aud.rspAggDMLActions @EventDate = '08/22/2008'

exec aud.rspAggDDLActions @EventDate = '08/22/2008'

These will populate the aud.rptAgg[AuditEvent]Actions tables with your summary statistics.

* It is suggested that you execute these stored procedures (e.g., by using SQL Server Agent) on a nightly basis to keep this daily information up to date. An example of a nightly job script would be:

-- Declare @LastDay variable to set it to last day date

-- the job will run after midnight to aggregate last day data

Declare @LastDay char(11)

select @LastDay = Convert(char(11), getdate()-1 , 1)

Select @LastDay

Exec aud.rspAggServerActions @LastDay

Exec aud.rspAggDatabaseActions @LastDay

Exec aud.rspAggDDLActions @LastDay

Exec aud.rspAggDMLActions @LastDay

**Which Partition Has Your Data?**

The SQL script to create your audit database (SQLAudit) will automatically create twelve monthly filegroups starting from the month you executed the script. As well, the partition scheme and functions will ensure that the aud.AuditLog\_% tables are partitioned by month and correlated to the monthly filegroups. To view which rows of data are in which table partitions, use the following script.

select partition\_id, OBJECT\_NAME(object\_id), object\_id, index\_id, partition\_number, partition\_id, rows as [RowCount], x.value

from sys.partitions

left outer join (

select boundary\_id, value

from sys.partition\_range\_values

where function\_id = (

select function\_id

from sys.partition\_functions

where [name] = 'monthly\_partition\_function'

)

) x

on x.boundary\_id = partition\_number - 1

where OBJECT\_NAME(object\_id) like 'AuditLog%' and index\_id = 1

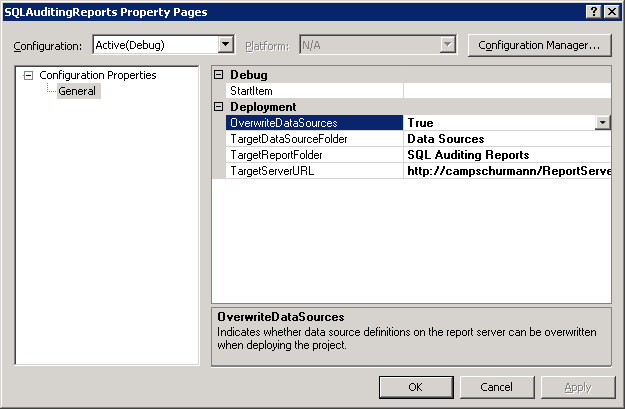
order by OBJECT\_NAME(object\_id), partition\_number

**View Your Reports**

As noted in the Auditing Sensitive Operations section above, there are a number of reports that can be generated by the audit logs. You can view these reports by deploying the SQLAuditReports Reporting Services solution.

1. Open the SQLAuditReports Reporting Services solution within the Microsoft Visual Studio® development system.
2. Change the Deployment TargetServerURL to your server, i.e., change <http://campschurmann/> to your server name.

**Figure 2:** SQL AuditingReports Property Page



1. Ensure your SQLAudit.rds shared data source is pointing to your own SQLAudit database (vs. some other server’s version of the SQLAudit database).
2. Deploy your reports (Build > Deploy) and you can view your reports at: http://[Your Server Name] /Reports/Pages/Folder.aspx?ItemPath=%2fSQL+Auditing+Reports&ViewMode=List

**Known Issues**

* There are times where the audit logs do not record the server instance name; i.e., the server instance name field is empty within the audit logs. To work around this, the SSIS package will go through the entire audit log and determine the server instance name and assign that name to the entire log (because an audit log comes from only one server). But there are situations where even that will not work, because the entire file has no server instance name. To mitigate this, set up your audit logs with the naming convention

SQLAudit$*%Server$InstanceName%*\_%GUID%.sqlaudit

[aud].[fn\_GetServerInstanceName] will parse the name of the audit file, pull out the highlighted %Server$InstanceName%, and use this name if one does not already exist. To name your log file with this naming convention, when you create your server audit, alter the name of your audit with the format of SQLAudit$Server$InstanceName.