mylifeismymessage.net

"AutoAudit" - CodePlex Free Utility to Build SQLServer Triggers

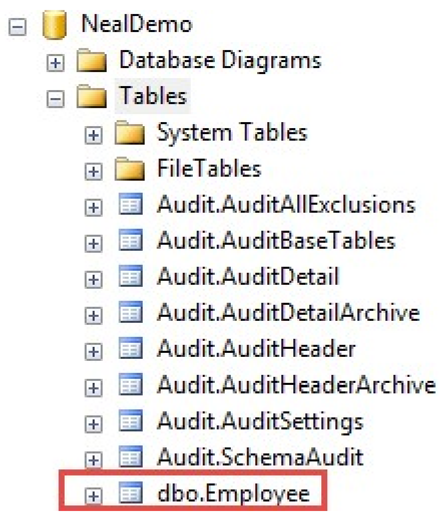
A few years ago I discovered a great utility called “Auto Audit” on CodePlex by Paul Nielson and John Sigouin. The best thing is that it creates triggers that audit insert, update, and deletes on any desired table. It creates a global audit table, and creates triggers specific to each of your tables. It can optionally add these columns to each table: CreatedDate, CreatedBy, ModifiedDate, ModifiedBy, and RowVersion (incrementing INT). It includes views to see your audits, and it does add eight of its own tables to your database (all nicely put in the ‘Audit’ schema). It also logs who is doing DDL commands, so you know, for example, which DBA or which developer, added some column to some schema.

It’s like a mini-framework to make sure all your triggers and auditing are done in a consistent manner, and it writes all the tedious trigger code for you. This of course allows you to see who changed what and when. The other benefit is that it standardizes the process, and builds the trigger for you. I’ve been in other companies where we had similar functionality, but it was all hand-coded, and rather tedious to create and update the triggers.

When you download it, what you get is just one big .sql file, 6422 lines long. You run that script in each database in which you want to use Auto Audit.

Installing

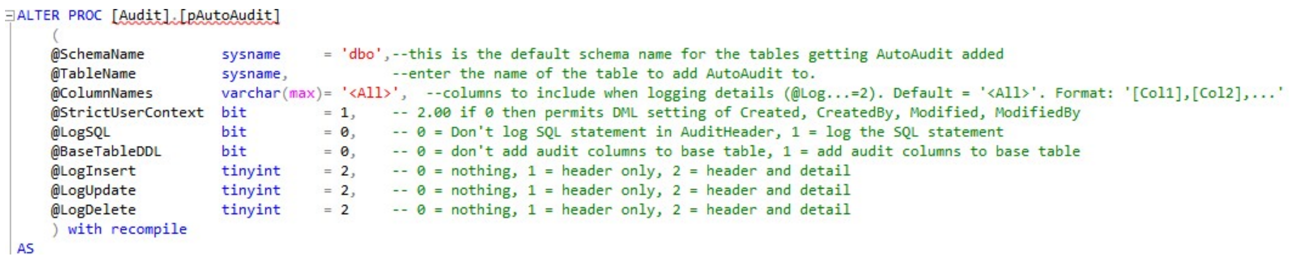
I had a database called NealDemo with one table dbo.Employee (in red box below). Running the Auto Audit script added 8 of it’s own tables, but as of yet, it does not touch the Employee table. We have to do the next step for that to happen.



Running the above will also create some stored procedures, views, and most importantly

SchemaAuditDDLTrigger DDL Trigger. This will allow you to see who for instance modified a table structure, because that will get audited too.

Turning On Audit for a Table



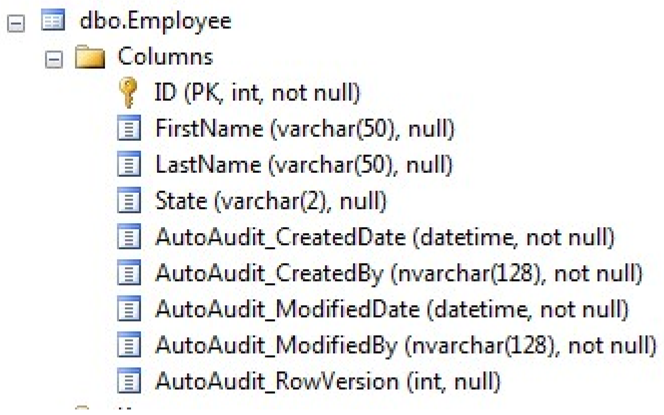
```
ALTER PROC [Audit].[pAutoAudit]
    (
    @SchemaName          sysname      = 'dbo',--this is the default schema name for the tables getting AutoAudit added
    @TableName           sysname,             --enter the name of the table to add AutoAudit to.
    @ColumnNames         varchar(max)= '<All>',  --columns to include when logging details (@Log...=2). Default = '<All>'. Format: '[Col1],[Col2],...'
    @StrictUserContext   bit          = 1,    -- 2.00 if 0 then permits DML setting of Created, CreatedBy, Modified, ModifiedBy
    @LogSQL              bit          = 0,    -- 0 = Don't log SQL statement in AuditHeader, 1 = log the SQL statement
    @BaseTableDDL        bit          = 0,    -- 0 = don't add audit columns to base table, 1 = add audit columns to base table
    @LogInsert           tinyint      = 2,    -- 0 = nothing, 1 = header only, 2 = header and detail
    @LogUpdate           tinyint      = 2,    -- 0 = nothing, 1 = header only, 2 = header and detail
    @LogDelete           tinyint      = 2     -- 0 = nothing, 1 = header only, 2 = header and detail
    ) with recompile
AS
```

So to turn on auditing for one single table, you do this (overriding the schema or other parms as desired):

exec [Audit].[pAutoAudit] @TableName='Employee', @BaseTableDDL=1

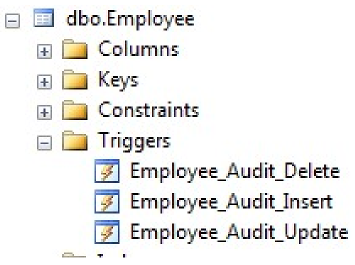
Note: If you don't specify @BaseTableDDL=1, the 5 columns won't get added to the table. Below is an example of my employee table, and it's fairly obvious which 5 columns were added.

If you are ready to turn it on everywhere, on all tables, instead use pAutoAuditAll.



```
dbo.Employee
  Columns
    ID (PK, int, not null)
    FirstName (varchar(50), null)
    LastName (varchar(50), null)
    State (varchar(2), null)
    AutoAudit_CreatedDate (datetime, not null)
    AutoAudit_CreatedBy (nvarchar(128), not null)
    AutoAudit_ModifiedDate (datetime, not null)
    AutoAudit_ModifiedBy (nvarchar(128), not null)
    AutoAudit_RowVersion (int, null)
```

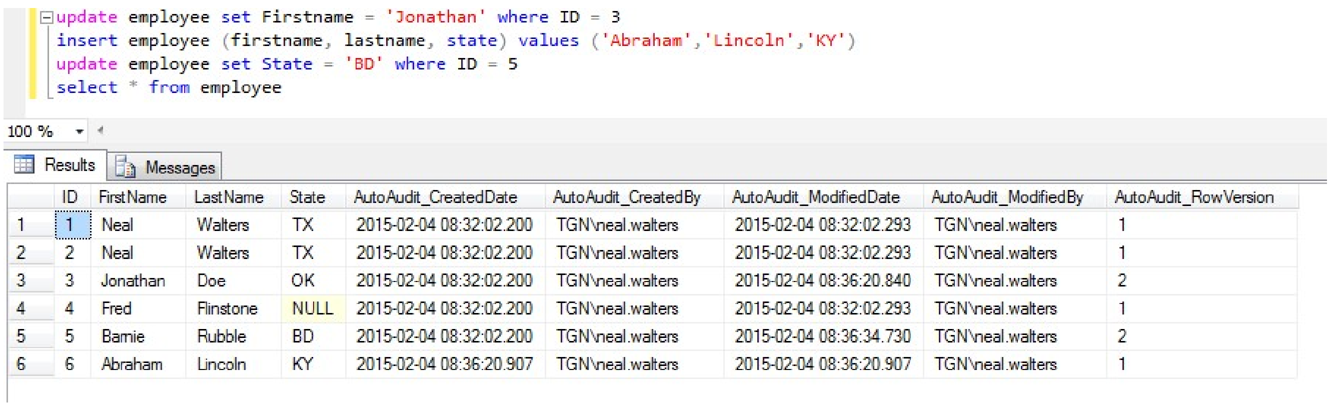
It created three triggers on my Employee table. The update trigger alone is 305 lines of code, and the other two are about 150 lines each.



```
dbo.Employee
  Columns
  Keys
  Constraints
  Triggers
    Employee_Audit_Delete
    Employee_Audit_Insert
    Employee_Audit_Update
```

Viewing the Audits

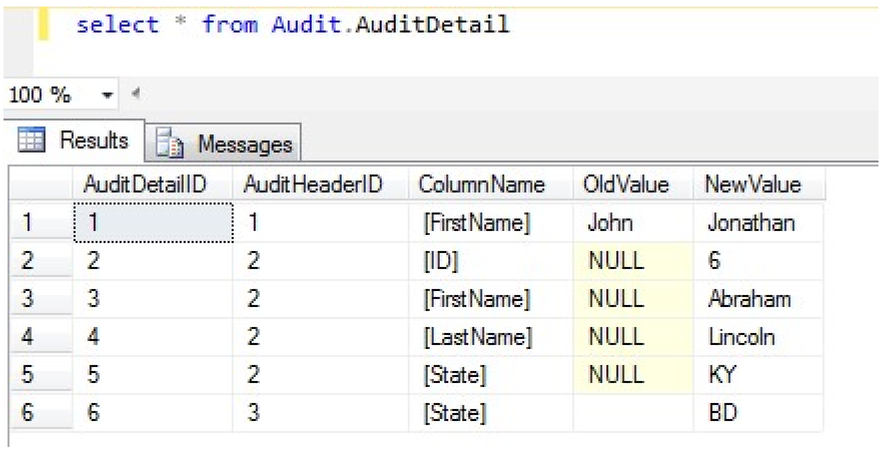
I ran the following update:



```
update employee set Firstname = 'Jonathan' where ID = 3
insert employee (firstname, lastname, state) values ('Abraham','Lincoln','KY')
update employee set State = 'BD' where ID = 5
select * from employee
```

	ID	FirstName	LastName	State	AutoAudit_CreatedDate	AutoAudit_CreatedBy	AutoAudit_ModifiedDate	AutoAudit_ModifiedBy	AutoAudit_RowVersion
1	1	Neal	Walters	TX	2015-02-04 08:32:02.200	TGN\neal.walters	2015-02-04 08:32:02.293	TGN\neal.walters	1
2	2	Neal	Walters	TX	2015-02-04 08:32:02.200	TGN\neal.walters	2015-02-04 08:32:02.293	TGN\neal.walters	1
3	3	Jonathan	Doe	OK	2015-02-04 08:32:02.200	TGN\neal.walters	2015-02-04 08:36:20.840	TGN\neal.walters	2
4	4	Fred	Flinstone	NULL	2015-02-04 08:32:02.200	TGN\neal.walters	2015-02-04 08:32:02.293	TGN\neal.walters	1
5	5	Barnie	Rubble	BD	2015-02-04 08:32:02.200	TGN\neal.walters	2015-02-04 08:36:34.730	TGN\neal.walters	2
6	6	Abraham	Lincoln	KY	2015-02-04 08:36:20.907	TGN\neal.walters	2015-02-04 08:36:20.907	TGN\neal.walters	1

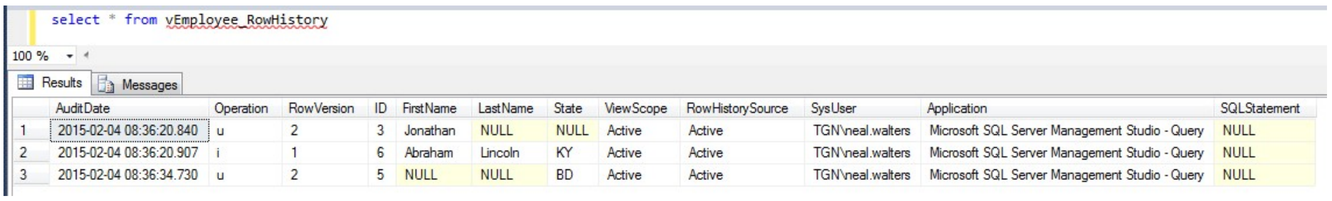
Rather than creating one audit table for each table, Auto Audit uses one consolidated audit table:



```
select * from Audit.AuditDetail
```

	AuditDetailID	AuditHeaderID	ColumnName	OldValue	NewValue
1	1	1	[FirstName]	John	Jonathan
2	2	2	[ID]	NULL	6
3	3	2	[FirstName]	NULL	Abraham
4	4	2	[LastName]	NULL	Lincoln
5	5	2	[State]	NULL	KY
6	6	3	[State]		BD

However, it does create a view for each table. In addition it creates a view called Employee_Deleted to show any rows deleted from my Employee table..



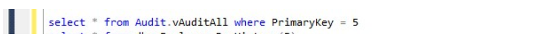
```
select * from vEmployee_RowHistory
```

	AuditDate	Operation	RowVersion	ID	FirstName	LastName	State	ViewScope	RowHistorySource	SysUser	Application	SQLStatement
1	2015-02-04 08:36:20.840	u	2	3	Jonathan	NULL	NULL	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL
2	2015-02-04 08:36:20.907	i	1	6	Abraham	Lincoln	KY	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL
3	2015-02-04 08:36:34.730	u	2	5	NULL	NULL	BD	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL

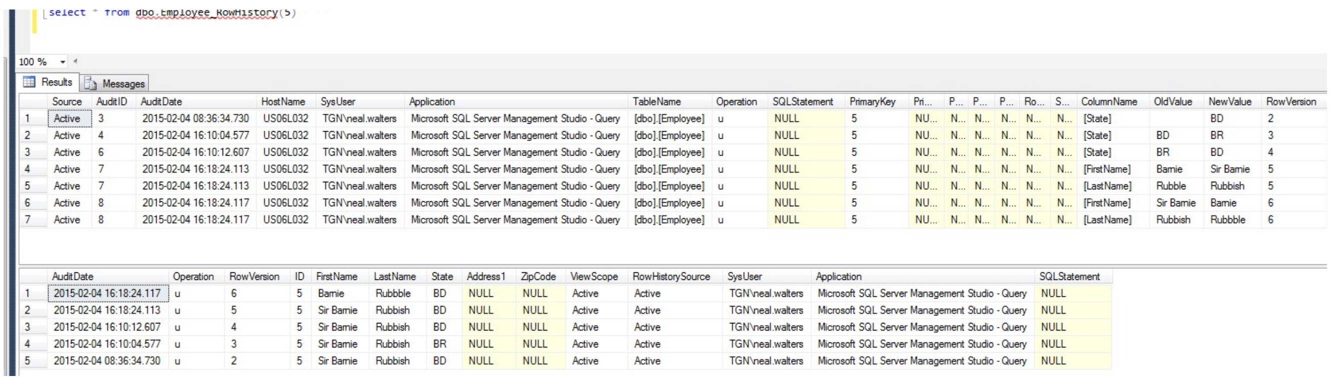
Now look what happens when you "Alter" a table. There is a Database Trigger called "SchemaAuditDDLTrigger". It catches the changes and logs them, as well as rebuilds the triggers on the Employee table.

Table Function

There's also a table function that can be used on a single key. Below I show the differnce in the vAuditAll and using the function against that same key:



```
select * from Audit.vAuditAll where PrimaryKey = 5
```



```
select * from dbo.Employee_RowHistory(5)
```

100 %

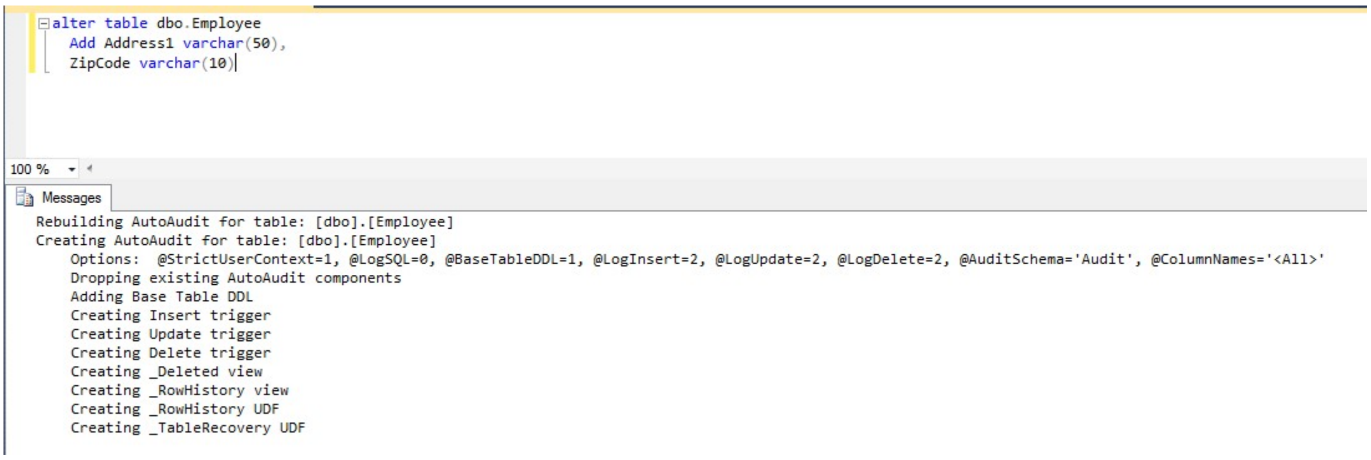
Results | Messages

	Source	AuditID	AuditDate	HostName	SysUser	Application	TableName	Operation	SQLStatement	PrimaryKey	Pri...	P...	P...	P...	Ro...	S...	ColumnName	OldValue	NewValue	RowVersion
1	Active	3	2015-02-04 08:36:34.730	US06L032	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	[dbo].[Employee]	u	NULL	5	NU...	N...	N...	N...	N...	N...	[State]		BD	2
2	Active	4	2015-02-04 16:10:04.577	US06L032	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	[dbo].[Employee]	u	NULL	5	NU...	N...	N...	N...	N...	N...	[State]	BD	BR	3
3	Active	6	2015-02-04 16:10:12.607	US06L032	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	[dbo].[Employee]	u	NULL	5	NU...	N...	N...	N...	N...	N...	[State]	BR	BD	4
4	Active	7	2015-02-04 16:18:24.113	US06L032	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	[dbo].[Employee]	u	NULL	5	NU...	N...	N...	N...	N...	N...	[FirstName]	Barnie	Sir Barnie	5
5	Active	7	2015-02-04 16:18:24.113	US06L032	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	[dbo].[Employee]	u	NULL	5	NU...	N...	N...	N...	N...	N...	[LastName]	Rubble	Rubbish	5
6	Active	8	2015-02-04 16:18:24.117	US06L032	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	[dbo].[Employee]	u	NULL	5	NU...	N...	N...	N...	N...	N...	[FirstName]	Sir Barnie	Barnie	6
7	Active	8	2015-02-04 16:18:24.117	US06L032	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	[dbo].[Employee]	u	NULL	5	NU...	N...	N...	N...	N...	N...	[LastName]	Rubbish	Rubbble	6

	AuditDate	Operation	RowVersion	ID	FirstName	LastName	State	Address1	ZipCode	ViewScope	RowHistorySource	SysUser	Application	SQLStatement
1	2015-02-04 16:18:24.117	u	6	5	Barnie	Rubbble	BD	NULL	NULL	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL
2	2015-02-04 16:18:24.113	u	5	5	Sir Barnie	Rubbish	BD	NULL	NULL	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL
3	2015-02-04 16:10:12.607	u	4	5	Sir Barnie	Rubbish	BD	NULL	NULL	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL
4	2015-02-04 16:10:04.577	u	3	5	Sir Barnie	Rubbish	BR	NULL	NULL	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL
5	2015-02-04 08:36:34.730	u	2	5	Sir Barnie	Rubbish	BD	NULL	NULL	Active	Active	TGN\neal.walters	Microsoft SQL Server Management Studio - Query	NULL

Auditing DDL Changes

Now look what happens when you "Alter" a table. There is a Database Trigger called "SchemaAuditDDLTrigger". It catches the changes and logs them, as well as rebuilds the triggers on the Employee table.



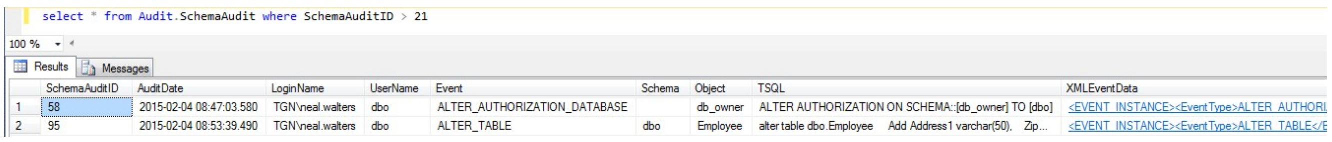
```
alter table dbo.Employee
  Add Address1 varchar(50),
  ZipCode varchar(10)
```

100 %

Messages

```
Rebuilding AutoAudit for table: [dbo].[Employee]
Creating AutoAudit for table: [dbo].[Employee]
    Options:  @StrictUserContext=1, @LogSQL=0, @BaseTableDDL=1, @LogInsert=2, @LogUpdate=2, @LogDelete=2, @AuditSchema='Audit', @ColumnNames='<All>'
    Dropping existing AutoAudit components
    Adding Base Table DDL
    Creating Insert trigger
    Creating Update trigger
    Creating Delete trigger
    Creating _Deleted view
    Creating _RowHistory view
    Creating _RowHistory UDF
    Creating _TableRecovery UDF
```

To view who changed the table or anything in the schema using DDL, check the Audit.SchemaAudit table:



```
select * from Audit.SchemaAudit where SchemaAuditID > 21
```

100 %

Results | Messages

	SchemaAuditID	AuditDate	LoginName	UserName	Event	Schema	Object	TSQL	XMLEventData
1	58	2015-02-04 08:47:03.580	TGN\neal.walters	dbo	ALTER_AUTHORIZATION_DATABASE		db_owner	ALTER AUTHORIZATION ON SCHEMA::[db_owner] TO [dbo]	<EVENT_INSTANCE><EventType>ALTER_AUTHORI
2	95	2015-02-04 08:53:39.490	TGN\neal.walters	dbo	ALTER_TABLE	dbo	Employee	alter table dbo.Employee Add Address1 varchar(50), Zip...	<EVENT_INSTANCE><EventType>ALTER_TABLE</E

Summary

I'm so impressed with this utility. If you have no auditing at all on your database, you can turn this on in just a matter of minutes. Note however, that it could impact your performance, so be sure to benchmark if you are pushing the boundaries on performance already.

Filed under: SQL