# CHANGE TRACKING FRAMEWORK for Microsoft SQL Server

Sergey Vaselenko

# CHANGE TRACKING FRAMEWORK for Microsoft SQL Server

Written by Sergey Vaselenko

This e-book describes the change tracking framework for Microsoft SQL Server.

You may add change tracking features to a database and track changes using SSMS or Microsoft Excel.

Also, you may extend the Microsoft Excel context menu to allow business users to track and rollback changes.

# Contents

Contents	1
Introduction	2
Chapter 1. Installation	3
Download	3
Download Package	3
Installation	3
Uninstalling	3
Chapter 2. Setup	4
Creating Administrator's Workbook	4
Creating Change Tracking Triggers	7
Useful Operations	9
Attaching Context Menus	9
Chapter 3. Usage	
Context Menu	
Task Panes	
Restoring Records	
ID Lookup	
Translation	14
Chapter 4. Permissions	15
Administrator Permissions	
User Permissions	
Chapter 5. Database Objects	
Roles	
Schemas	
Tables	
Views	
Procedures	
Triggers	
Conclusion	
About the Author	23

## Introduction

The change tracking framework provides features that track DML changes (insert, update, and delete operations) in a database.

You may:

- Create and drop change tracking triggers.
- Create and drop procedures to select changes.
- Restore changed records.

You may execute the operations in SSMS, your application, or in Microsoft Excel using the SaveToDB add-in.

Moreover, you may easily extend Excel context menu to track changes in Excel.

The framework includes the following blocks:

- Database tables, views, and procedures to store and manage changes;
- Configuration objects to manage change tracking in Microsoft Excel for administrators;
- Configuration objects to track and restore changes in Microsoft Excel for end users.

You may download and install the framework in a couple of minutes.

The framework uses features of the SaveToDB Express add-in for Microsoft Excel. So, you may use it for free.

Microsoft SQL Server includes built-in features to track changes. You may read the following article, for example:

Track Data Changes (SQL Server)

https://docs.microsoft.com/en-us/sql/relational-databases/track-changes/track-data-changes-sql-server

We have created the change tracking framework for SQL Server to get the following features:

- 1. To see historical data in Microsoft SQL Express (not supported by the Change Data Capture feature).
- 2. To easily integrate and use the solution in Microsoft Excel (not supported by SQL Server).
- 3. To allow business users to manage change tracking features and rollback changes using Microsoft Excel.

We get a reliable and high-performance solution that you may use for free.

Try the framework. You may save a lot of time and make your users happier.

Best regards, Sergey Vaselenko November 8, 2017

# Chapter 1. Installation

#### Download

You may download the change tracking framework at <u>https://www.savetodb.com/download.htm</u>.

The application also requires SaveToDB 7.14 or higher. You may download the add-in here too.

#### **Download Package**

Unzip the downloaded package to a local drive.

The download package contains the ChangeTrackingFramework folder with the following files:

- change-tracking-framework-install.sql
- change-tracking-framework-remove.sql
- change-tracking-framework-for-sql-server.pdf

#### Installation

To install the change tracking framework using SQL Server Management Studio (SSMS), open and execute (F5) the **change-tracking-framework-install.sql** script. The script creates the following framework objects:

 □
 Tables (filtered)
 □
 Views (filtered)
 □

 □
 System Tables
 □
 System Views
 □

 □
 FileTables
 □
 logs.view\_column\_translations

 □
 Iogs.base\_tables
 □
 logs.view\_event\_handlers

 □
 Iogs.change\_logs
 □
 Iogs.view\_object\_translations

 □
 Iogs.column\_translations
 □
 Iogs.view\_objects

 □
 Iogs.event\_handlers
 □
 Iogs.view\_query\_list

 □
 Iogs.object\_translations
 □
 Iogs.view\_query\_list

- Stored Procedures (filtered)
  - System Stored Procedures

  - 🕀 📃 logs.usp\_create\_triggers
  - ∃ logs.usp\_drop\_all\_triggers
  - 🕀 📃 logs.usp\_drop\_triggers
  - Iogs.usp\_restore\_current\_record

  - 🕀 🧾 logs.usp\_restore\_record

  - Iogs.usp\_select\_record
  - Iogs.usp\_select\_records
  - Iogs.usp\_set\_role\_permissions\_admin
  - Iogs.usp\_set\_role\_permissions\_users

#### Uninstalling

To uninstall the framework, open and execute the **change-tracking-framework-remove.sql** script.

Note that the script removes frameworks objects, created change tracking triggers and procedures.

# Chapter 2. Setup

### Creating Administrator's Workbook

You may use Microsoft Excel to manage change tracking framework objects.

Create a new workbook and run Wizards, Data Connection Wizard:

E	5-0	,⇒ - ∓			囨	—		×			
File	e Ho	me Ins	ert Pag	e Layout	Formulas	Data	View	SaveTo	DB 🤇	? Tell me	Ŀ
Save	Reload	Active Que	ery: I Query List		Table View:	×	Wizards	Options	Help		
Table	e Data 🛛 🕞	(	Query List	G	Table Vie	ws G	🐻 Da	ata Connec	ction Wiz	ard	
	A	В	С	D	E	F	Pi	vot Data C	onnectio	on Wizard	

Select the first data provider to connect to Microsoft SQL Server and click Next:

Data Connection	🚺 Data Connection Wizard ? 🗙											
Select Provider												
Select an appropriat	e provider to	connect to a data source										
Server	Туре	Description		^								
SQL Server/SQL Azure	OLEDB	Microsoft OLE DB Provider for SQL Server										
SQL Server/SQL Azure	OLEDB	SQL Server Native Client 11.0										
SQL Server/SQL Azure	ODBC	SQL Server										
SQL Server/SQL Azure	ODBC	SQL Server Native Client 11.0										
SQL Server/SQL Azure	ODBC	ODBC Driver 13 for SQL Server										
SQL Server/SQL Azure	ODBC DSN	CRM										
Oracle Database	.NET	Oracle Data Provider for .NET										
IBM DB2	.NET	IBM DB2 .NET Data Provider 10.5.0										
IBM DB2	ODBC	IBM DB2 ODBC Driver										
MySQL	.NET	MySQL Data Provider		$\checkmark$								
This provider is preferable	e to connect t	o Microsoft SQL Server and Windows Azure SQL Dat	tabase.	~								
				~								
Help		Cancel Back Next	Finish									

Select the logs.view\_query\_list in the Query List, uncheck the checkbox as shown, and select logs.view\_objects:

Data Connection	Wizard				?	×					
Select Object	nd the object to co	onnect									
Select Query List:											
logs.view_query_list		~	Enable Query	List on the ri	bbon	_					
Select the query object	t:	_									
Schema 🗸	Name	Туре	<ul> <li>Trans</li> </ul>	ated Name		Des					
logs	base_tables	base table	logs.ba	se_tables							
logs	view_objects	view	logs.vie	ew_objects							
Help S	QL	Cancel	Back	Next	Fin	ish					

In the next screen, select the **schema** and **has\_log\_triggers** field to place to the ribbon to filter data and click **OK**:

	Query Parameters -	logs.view_objects	;	?	×						
Sele	select columns to place them to the Ribbon and to filter the data										
	Parameter	DataType	Value		w						
$\checkmark$	object_id	int		-	🗆						
$\checkmark$	schema	nvarchar(128)		-	🗹						
$\checkmark$	name	nvarchar(128)		-							
$\checkmark$	insert_trigger	nvarchar(128)		-							
$\checkmark$	update_trigger	nvarchar(128)		-							
$\checkmark$	delete_trigger	nvarchar(128)		-							
$\checkmark$	select_handler	nvarchar(128)		-							
$\checkmark$	has_log_triggers	int		-	🗹						
	Help OK Cancel										

Select cell **B3** to insert a table and click **OK**.

Input		?	×
Select a cell to inser \$B\$3	t a query		
C	OK	Ca	ncel

The SaveToDB add-in inserts the table. You see the database objects including columns to display triggers:

H	<b>গ</b> - ∂-	÷			Book1 - Excel							
File	Home	Inser	Page Layout	Form	ulas Data	View	SaveTo	DB	♀ Tell n	ne what you		
Save	Reload Actio	Actions	ve Query: s.view_objects Reload Query List	Ŧ	schema: has_log_triggers		-	v v	Table View			
Ta	able Data	Es l	Query List	E.	Query	Parameter	'S	E.	Table	/iews		
A 🔊	С	D	E		F	6	;		Н			
1												
2												
3	object_id 🔻	schema	▼ name	<b>–</b> i	nsert_trigger 💌	update_t	rigger 💌	delet	te_trigger	select_ha		
4	1487134	dbo25	units									
5	129487590	dbo25	dimensions					1				
6	257488046	dbo25	form_permissions									
7	305488217	dbo25	dimension_calc_ty	pes				1				
4	Þ	Sheet1	( + )						:			
Ready												

Format the table as you like and save the workbook:

E	3	ڻ- ر <del>ه</del>		log_admin.	klsx - Excel		Ē	- 0	×
Fi	le	Hom	e Insert Page La	ayout Formula	is Data	Vie	w SaveToDB	♀ Tell me	Ŕ
	Α	D	E	F	G		Н	1	<b>▲</b>
1									
2				_		_			_
3		schema 💌	name 💌	insert_trigger 💌	update_trigg	er 🔻	delete_trigger 💌	select_handler	<b>–</b> –
4		dbo25	units						
5		dbo25	dimensions						
6		dbo25	form_permissions						
7		dbo25	dimension_calc_types						
8		dbo25	members						
9		dbo25	factors						
10		dbo25	facts						
11		dbo25	form_dimensions						
12		dho25	form rows						-
	0	Þ	log_admin (+	)	:	•			Þ
Rea	dy	<b>a</b>				Ξ	── -	+	90 %

#### **Creating Change Tracking Triggers**

The framework creates three triggers to log INSERT, UPDATE, and DELETE operations for captured tables.

Use the context menu to create and drop triggers:

E	3		~ <b>-</b>		log_admin.xlsx - Excel							-		×
Fi	ile	Home	e Insert	Page La	iyout	Formulas	Data	Vie	w Save	ToDB	Design	Q 1	Tell me	Ŀ
	Α	D		E		F		(	G		Н		1	
1														
2												_		_
3		schema 💌	name		-	insert_trigger	👻 upo	late_t	rigger 💌	delete	_trigger	selec	t_handler	· • 🗆
4		dbo25	units			44				1				
5		dbo25	dimensions	s Cali	ori 🔻	A 11	A S	- 9	6 ' ∺					
6		dbo25	form_perm	is: B	$I \equiv$	E 🕭 - 🗛 -	····· +	). 0.→ ← 00.	0 💉					
7		dbo25	dimension_	_carc_ty	pes [									
8		dbo25	members		- T		1							
9		dbo25	factors		Save	I ODB Drill-Dov	vn	P	Cre	ate Trig	)gers			
10		dbo25	facts	- X	Cut				Dro	p Trigg	ers			
11		dbo25	form_dime	ns 💼	Conv	,			Cla	arlogs				
12		dho25	form rows		<u>_</u> opy				CIE	ar Logs				
	•	- F	log_admi	n D	Paste	e Options:								Þ
Rea	dy				Ĉ							-	+	90 %

You have to set 1 into the **execute\_script** field to confirm creating triggers:

🔋 Query Param	Query Parameters - Create Triggers										
Edit procedure pa	Edit procedure parameters										
Parameter	DataType	Value				w					
schema	nvarchar(128)	dbo25			•	🗹					
name	nvarchar(128)	members			•	🗹					
execute_script	execute_script bit 1										
Help	Help OK										

Check the target table and click **OK**.

The SaveToDB add-in updates the table, and you see actual triggers for the captured tables:

E	5-	¢~ ∓		log_admin.xlsx	: - Excel			Ŧ	-		×
Fil	le H	ome Insert	Page Layout	Formulas	Data Viev	v Save	ToDB	Design	ŢΩ	ell me	Ŀ
	A D		E	F	(	6		Н		1	
1											
2					_				_		
3	schem	a 🔻 name	-	insert_trigger	update_t	rigger 🔻	delete	_trigger 💌	select	_handler	<b>-</b>
4	dbo25	units									_
5	dbo25	dimensi	ons								
6	dbo25	form_pe	rmissions								
7	dbo25	dimensi	on_calc_types								
8	dbo25	member	s	trigger_member	rs_trigger_m	embers_	l trigger	_members	xl_log	_membe	rs
9	dbo25	factors									
10	dbo25	facts									
11	dbo25	form_di	mensions								
12	dho25	form ro	M/S								
-		log_ad	min (+	)		-					Þ
Read	dy 🔝				E			-		+	90 %

In this example, we have created triggers for the dbo25.members table. Its triggers look like:



The change tracking framework also creates a stored procedure to select changes.

The procedures have the xl\_log\_ prefix. You may use filters in SSMS to filter change-tracking objects:

Stored Procedures (filtered)
 System Stored Procedures

Note that the framework creates procedures in the schemas of underlying tables.

So, if a user has SELECT and EXECUTE permissions for a schema, he can execute a new procedure.

The framework configures such procedures as SaveToDB context menu items for underlying tables.

#### **Useful Operations**

You may use the Actions menu to drop all triggers at once and the ribbon parameters to filter objects:

H	R S → C → ∓ log_admin.xlsx - Excel														
File	Ho	me	Insert	Page	Layout	Formu	las D	ata	View	SaveToDB	Q	Tell me what	you		
		la 🥅	Active	Query:			schema:			-	Tabl	e View:			
1000 Carrier	L <b>S</b>			logs.view_obje		iew_obje	cts	-	has_log_triggers:		<b>.</b>				-
- Jave	*	Actions ▼	là Re	load Que	ry List							💷 📭 🔛			
Ta	ble Data	Dr	rop All Tr	riggers	st	Es.		Query P	aramete	rs r	- I	Table Views			
A 🔊	D				-	F		G		Н		1			
1															

#### **Attaching Context Menus**

The framework configures procedures to show changes as SaveToDB context menu items for underlying tables.

To attach context menu items to other objects, edit the link data in the logs.base\_tables table.

You may do this in Excel. Just connect to the table, edit the data and click the **Save** button.

日	<b>€</b> - (	,∛ - ∓			log_i	admin.xlsx	- Exe	cel			Ŧ		-		×	<
File	Но	me	nsert	Page Layout	F	ormulas	Da	ata	Viev	N	SaveTo	DB	Q	Tell me	Ľ	ŝ
Save	Reload	Active C logs.ba	Query: se_tables pad Query	List	Tab	ole View:	**	Ŧ	Wiza	rds (	Dptions	He	D Ip			
Table [	Data 🗔		Query L	st 5	i I	Table View	/S	G.	(	Optio	ons	He	lp			^
1 2 3		RIECT SC			E v	BASE TAB		СНЕМ		BAS		NAN		1		
4	1 xls	s25		usp_members	5	dbo25		GHEM		men	nbers	_NAN				
6 7																Ŧ
-	Þ	log	admin	base_table	s	(+)		:	•						Þ	
Ready									E	Ε	J			+	90 %	5

In this example, we have attached **dbo25.members** context menu items to the **xls25.usp\_members** procedure.

As a result, users may track changes using the context menu of the **xls25.usp\_members** procedure.

They need click **Reload**, **Reload Data and Configuration** to load new context menu items.

Note that the target object must select all primary key fields of the base table with the same names.

# Chapter 3. Usage

#### **Context Menu**

The framework creates procedures to select changes of specific rows or entire tables by change types:

日	∙ ج		≓ Bo	ook2 -	Excel				Ŧ	—		×
File	Но	ome	Insert Page Layout Forn	nulas	Data	View	Save	ToDB	Design	Qт	ell me	Ŀ
A	С	Cali	ibri - 11 - 🕰 🗸 🖓 -	%,					G			H 🔺
2		B		.00	·							
3	id 🔻 d	men	ISION_IASOTL_OTAET ~ COAE	- <b>→</b> .0 <b>\</b>			v	name			comn	nent
4	1		SaveTeDP Drill Dewn		0			Accou	nts			
5	2		Saverous Unit-Down		Chang	ge Log		TIMES				
6	3	<b>ж</b>	Cu <u>t</u>		All Ch	anges		CATEG	ORIES			
7	4	E.	Сору		Incorte	- ad Rows		ENTITI	ES			
8	5				inseru	eu nows	. Г	DIM5				
9	6	ß	Paste Options:		Updat	ed Rows	- E	DIM6				
10	7	1	Ê		Delete	d Rows		DIM7				
11	8	1		-				Years*				
12	9		Paste Special	rs*				Quarte	ers*			
4	÷.	Ĵ	Smart <u>L</u> ookup				4					
Ready		ß	<u>R</u> efresh				Ξ		-	:	+	90 %

Click **Reload**, **Reload Data and Configuration** to refresh the context menu after creating the procedures.

#### **Task Panes**

The SaveToDB add-in shows changes in the Excel task panes. Initially, the log can be empty:

E	ਜ਼ ਨਾਟਾ ਦ					Book2 ·	Table Tools				
F	ile		Home	Insert	Page Layo	out Formulas	Data	Vie	w SaveToDB	Design	© Tell
	Α	С	[	D	E	F	:		G		H
1											
3		id 🔻	dimensio	on_id 🔻	sort_order 💌	code		-	name	▼ com	ment 🔻 i
4		1		1	1	ACCOUNTS			Accounts		
5		2		2	1	TIMES			TIMES		
6			1			1				I	
7		-	Char	nge Lo	g						
9			id		dimension_id	sort_order	code	2	name	comr	nent
10						Columns					
11						Columns			Id		
12						Reload		•	dimension_id		
13		1						- ·	sort_order		

You may configure columns shown in the task pane. For example, you may leave the most useful columns:

Change L	og			- ×
id	name	Change Action	Change Date	Change User

Let's change the name of the Accounts member and save the changes.

B	•5 •		Ŧ			Book2 -	Excel				Ŧ	-	_		>	<
File	н	ome l	nsert	Page Layout	Fo	rmulas	Data	View	Save	ToDB	Desi	ign	Ωте	ell me	Ľ	Ż
Save Table	Reloa Data	d Active d dbo2	e Query: 5.meml eload Q Que	oers uery List ery List		Table	View: I 📳 🔛 able Views		Wiz	ards O Optior	ptions	Help Help				~
A 🔊	C	D	)	E			F				(	G			Н	
1 2										_						
3	id 🔻 d	limensio	n_id _∽	sort_order	r CO	de			-	name			-	com	ment	
4	1		1		1 AC	COUNTS	;			Accou	nts2					
5	2		2	2	1 TI	MES				TIMES				<u> </u>		
6	3		3	4	1 CA	TEGORI	ES			CATEG	ORIES					
7	4		4	L	1 EN	TITIES				ENTIT	IES					•
-	×.	Sh	eet1	$(\pm)$					•						Þ	]
Ready									Ξ			-	+	+	90 %	6

Click the **Reload** menu item in the Change Log task pane (or run it from the context menu once again):

Change L	og			- ×
id	name	Change Action	Change Date	Change User
1	Accounts2	updated	07.11.2017 13:32	dbo

You see the changes.

The SaveToDB add-in also opens a new task pane to show record details (non-empty values):

Row Changes	- ×
Name	Value
id	1
dimension_id	1
sort_order	1
code	ACCOUNTS
name	Accounts2 [Accounts]
calc_type_id	0
row_color	11
is_active	1
Change Action	updated
Change Date	Nov 7 2017 1:32PM
Change User	dbo

You may see old values in square brackets.

#### **Restoring Records**

The context menu of the log records contains more items:

Change L	og				-	×
id	name	Change Act	ion	Change Date	Change Use	er
1	Accounts2	updated		Format Alignment Auto Size Columns Columns Reload	dho } 5	
				Restore Current Re	cord	
				Restore Previous R	ecord	

You may restore the current or previous version of the active row.

In this example, we want to restore the previous name. So, choose the previous record.

Query Parameters - Restore Previous Record         ?										
Edit procedure	Edit procedure parameters									
Parameter	DataType	Value		W						
change_id	int	1	-	🗹						
confirm	bit	0	-	🗹						
Help	]	ОК	С	ancel						

You must set 1 in the **confirm** field (to prevent unwanted changes) and click **OK**:

The add-in executes the **logs.usp\_restore\_record** procedure to restore changes and updates data:

Change Log									
id	name	Change Action	Change Date	Change User					
1	Accounts2	updated	07.11.2017 13:32	dbo					
1	Accounts	updated	07.11.2017 13:42	dbo					

#### ID Lookup

The task pane shows table rows as is.

Use the **ID Lookup** context menu item to find the related row in the foreign key table.

For example, this screen shows a dimension row of the account member:

ID Lookup			
id	code	name	parameter_name
1	accounts	Accounts	Account

#### Translation

The change tracking framework supports translation. Select the UI and data languages in the SaveToDB Options:

SaveToDB Option	s		?	$\times$
Common Options	Excel Options	Ribbon Options	Developer Options	
Languages				
Interface langua	age	French		~
Default data lar	iguage	French		~
This workbook	data language	The same as o	lefault	~
Password End	ryption			
Encrypt cor	nection string p	asswords by defa	ault	
Encrypt cor	nection string p	asswords in this	workbook	
Database Opt	ions			
Connection tim	eout, sec:			15
Command time	out, sec:			60
Help			OK Car	ncel

To apply changes, click **Reload**, **Reload Data and Configuration** and restart Excel. Voilà:

Modifier le journal							r x	, ,
id		name		Action de mod	Date de modif	Utilisa	teur de	
	1	Accounts2		actualisé	07.11.2017 13:32	dbo		
	1	Accounts		Format  Alignement Colonnes de taille automatique Colonnes			1	
				Rafraîchir				
				Recherche d'identifiant				
				Restaurer l'enregistrement en cours				
				Restaurer l'enregistrement précédent				

#### **Administrator Permissions**

Users of the log\_administrators role have all user permissions and additionally can:

- Create and drop change tracking triggers;
- Clear logs.

To create and drop triggers, users must also have such permissions in the target schema.

To clear logs, users must also have UPDATE permissions on the captured tables.

You may find log\_administrators permissions in the logs.usp\_set\_role\_permissions\_administrators procedure.

#### **User Permissions**

Users of the log\_users role can:

- Select log records;
- Restore records from the log.

To select log records, users must also have SELECT permissions on the captured tables.

To restore records, the user must also have UPDATE permissions on the captured tables.

You may find log\_users permissions in the logs.usp\_set\_role\_permissions\_users procedure.

You may use the **sp\_addrolemember** procedure to assign a role to a user:

EXEC sp\_addrolemember 'log\_users', 'pa\_user\_01'

#### Roles

The change tracking framework creates the following roles:

- log\_administrators
- log\_users

#### log\_administrators

Assign this role to users who can create triggers, drop triggers and clear logs.

See actual database permissions in the logs.usp\_set\_role\_permissions\_administrators procedure.

#### log\_users

Assign this role to business users who will use the change tracking functions.

See actual database permissions in the logs.usp\_set\_role\_permissions\_users procedure.

#### **Schemas**

#### logs

The change tracking framework creates its objects in the logs schema.

#### **Tables**

The change tracking framework contains the following tables in the logs schema:

- base\_tables
- change\_logs
- column\_translations
- event\_handlers
- object\_translations

#### base\_tables

This user table contains a configuration used to attach context menu items to database objects.

See Attaching Context Menus.

#### change\_logs

This application table contains change tracking data for all captured tables. Do not edit it.

The table contains primary keys and both current and previous row values of source records in an XML format.

#### column\_translations

This application table contains column translation data. You may change and add translations.

#### event\_handlers

This application table contains event handler configuration.

The framework inserts and deletes handlers of every captured table here.

Do not edit it.

#### object\_translations

This application table contains object translation data. You may change and add translations.

#### Views

The change tracking framework contains the following views in the logs schema:

- view\_column\_translations
- view\_event\_handlers
- view\_object\_translations
- view\_objects
- view\_query\_list

#### view\_column\_translations

This view selects column translation configuration for the SaveToDB add-in.

#### view\_event\_handlers

This view selects event handler configuration for the SaveToDB add-in.

#### view\_object\_translations

This view selects object translation configuration for the SaveToDB add-in.

#### view\_objects

This view selects database objects and related change tracking framework information.

Use it to configure change tracking triggers. See <u>Setup</u>.

#### view\_query\_list

This view selects change tracking framework objects to connect with Microsoft Excel.

See Creating Administrator's Workbook.

#### **Procedures**

The change tracking framework contains the following views in the logs schema:

- usp\_clear\_logs
- usp\_create\_triggers
- usp\_drop\_all\_triggers
- usp\_drop\_triggers
- usp\_restore\_current\_record
- usp\_restore\_previous\_record
- usp\_restore\_record
- usp\_select\_lookup\_id
- usp\_select\_record
- usp\_select\_records
- usp\_set\_role\_permissions\_administrators
- usp\_set\_role\_permissions\_users

#### usp\_clear\_logs

This procedure clears change tracking records of the specified target table.

A user must also have the UPDATE permission on the target table.

You may run it from the context menu of the logs.view\_objects view.

See Creating Change Tracking Triggers.

#### usp\_create\_triggers

This procedure creates change tracking triggers of the specified target table.

A user must also have the ALTER permission on the target table.

You may run it from the context menu of the logs.view\_objects view.

See Creating Change Tracking Triggers and Triggers.

#### usp\_drop\_all\_triggers

This procedure drops all change tracking triggers and clears all logs. A user must also have the ALTER permission on altered tables. You may run it from the Actions menu of the logs.view\_objects view. See <u>Useful Operations</u>.

#### usp\_drop\_triggers

This procedure drops change tracking triggers of the specified target table. A user must also have the ALTER permission on the target table. You may run it from the context menu of the logs.view\_objects view. See Creating Change Tracking Triggers.

#### usp\_restore\_current\_record

This procedure restores the current version of the change log record.

#### See Restoring Records.

The procedure executes the usp\_restore\_record procedure with the predefined parameter.

#### usp\_restore\_previous\_record

This procedure restores the previous version of the change log record.

See Restoring Records.

The procedure executes the usp\_restore\_record procedure with the predefined parameter.

#### usp\_restore\_record

This procedure restores a row from the change log record.

A user must have the UPDATE permission on the captured table.

#### usp\_select\_lookup\_id

This procedure selects rows from tables referenced by foreign keys.

See ID Lookup.

A user must have the SELECT permission on the captured and referenced tables.

#### usp\_select\_record

This procedure selects change details of a change log record.

See Task Panes.

A user must have the SELECT permission on the captured table.

#### usp\_select\_records

This procedure selects change log records.

Generated procedures like dbo25.xl\_log\_members execute this procedure to select changes.

See Creating Change Tracking Triggers and Task Panes.

A user must have the SELECT permission on the captured table.

#### usp\_set\_role\_permissions\_administrators

This procedure sets permissions for the log\_administrators role. See Administrator Permissions.

#### usp\_set\_role\_permissions\_users

This procedure sets permissions for the log\_users role. See User Permissions.

#### Triggers

The logs.usp create triggers procedure generates three triggers for every captured table like these:

- trigger\_members\_log\_insert
- trigger\_members\_log\_update
- trigger\_members\_log\_delete

Below are the trigger examples.

#### **Insert Trigger**

```
CREATE TRIGGER [dbo25].[trigger_members_log_insert]
    ON [dbo25].[members]
    AFTER INSERT
AS
BEGIN
SET NOCOUNT ON
INSERT INTO logs.change_logs
   (object_id, id, inserted, deleted, change_type, change_date, change_user)
SELECT
    369488445
    , inserted.[id]
    , (SELECT * FROM inserted FOR XML RAW)
    , NULL
    , 1
    , GETDATE()
    , USER_NAME()
FROM
    inserted
END
```

#### **Update Trigger**

```
CREATE TRIGGER [dbo25].[trigger_members_log_update]
    ON [dbo25].[members]
    AFTER UPDATE
AS
BEGIN
SET NOCOUNT ON
INSERT INTO logs.change_logs
   (object_id, id, inserted, deleted, change_type, change_date, change_user)
SELECT
    369488445
    , inserted.[id]
    , (SELECT * FROM inserted FOR XML RAW)
    , (SELECT * FROM deleted FOR XML RAW)
    , 3
    , GETDATE()
    , USER_NAME()
FROM
    inserted
END
```

#### **Delete Trigger**

```
CREATE TRIGGER [dbo25].[trigger_members_log_delete]
    ON [dbo25].[members]
    AFTER DELETE
AS
BEGIN
SET NOCOUNT ON
INSERT INTO logs.change_logs
   (object_id, id, inserted, deleted, change_type, change_date, change_user)
SELECT
    369488445
    , deleted.[id]
    , NULL
    , (SELECT * FROM deleted FOR XML RAW)
    , 2
    , GETDATE()
     USER_NAME()
FROM
    deleted
END
```

#### **Change Log Table**

Triggers save records to the logs.change\_logs table that has the following declaration:

```
CREATE TABLE [logs].[change_logs] (
       [change_id] int IDENTITY(1,1) NOT NULL
    , [object_id] int NOT NULL
    , [id] int NULL
    , [keys] nvarchar(445) NULL
    , [keys] nvarchar(445) NULL
    , [deleted] xml NULL
    , [deleted] xml NULL
    , [change_type] tinyint NOT NULL
    , [change_date] datetime NOT NULL
    , [change_user] nvarchar(128) NOT NULL
    , [CONSTRAINT [PK_change_logs_logs] PRIMARY KEY ([change_id])
)
```

#### Comments

All triggers of all tables add new change tracking records to the single logs.change\_logs table.

If a table has an identity field, the triggers use this field. This is the fastest case to track and select changes.

Otherwise, the triggers pack the key field values into XML. The maximum length is 445 characters.

You cannot use the framework for tables with larger key lengths.

Triggers pack inserted and deleted values into XML.

Also, triggers save the time and user of the changes.

# Conclusion

To add change tracking features to an example database, we have made the following steps:

- 1. Install the change tracking framework.
- 2. Create administrator's workbook.
- 3. Create triggers for captured tables.

You may repeat these steps for any database in a couple of minutes.

However, you get a reliable and high-performance solution. For free.

Moreover, you get the possibility to work with logs in Microsoft Excel using the free SaveToDB Express add-in.

And, you may extend your existing Microsoft Excel application to allow users track changes and restore records.

I hope you try it and will like it.

Best regards,

Sergey Vaselenko

# About the Author



My name is Sergey Vaselenko.

I am from Russia, Moscow.

My passion is creating software.

I am a founder and CEO of Gartle Technology Corporation and a leading developer of the SaveToDB add-in.

You are welcome to contact me at www.facebook.com/sergey.vaselenko www.linkedin.com/in/vaselenko/