How to setup SQL Server tracing using Extended Events for Sage X3 v7+

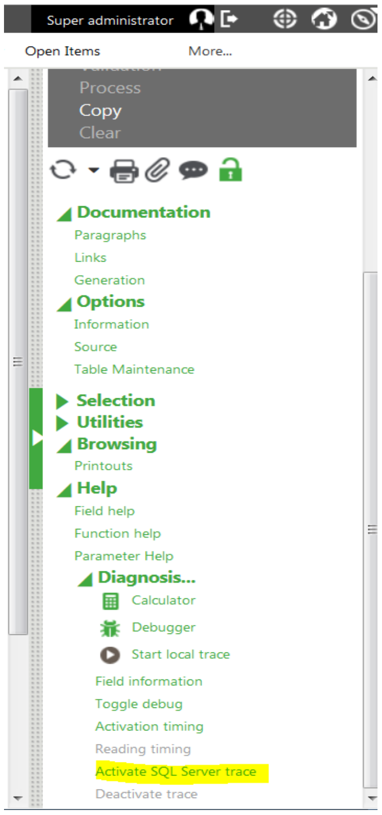
Disclaimer

Document Information

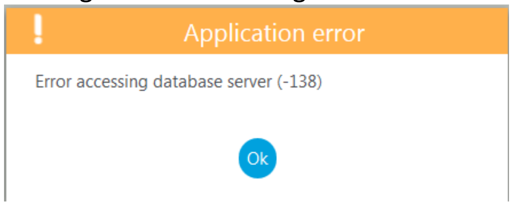
Author: Mike Shaw, Sage UK X3 Support Team

Introduction

Ordinarily you can enable SQL Server tracing from within an X3 User session, by navigating to Help→ Diagnostics and then selecting “Activate SQL Server trace” as shown below



There are several reasons why SQL tracing may not be able to be launched using this method. For example, if your database is on a different server to the X3 Process server, then you will see an error message “Error accessing database server (-138)” as below:



Error accessing database server (-138)

The simplest way to overcome this message is to manually identify and trace the X3 user session from within SQL Server Management Studio, as described in the main section of this document below.

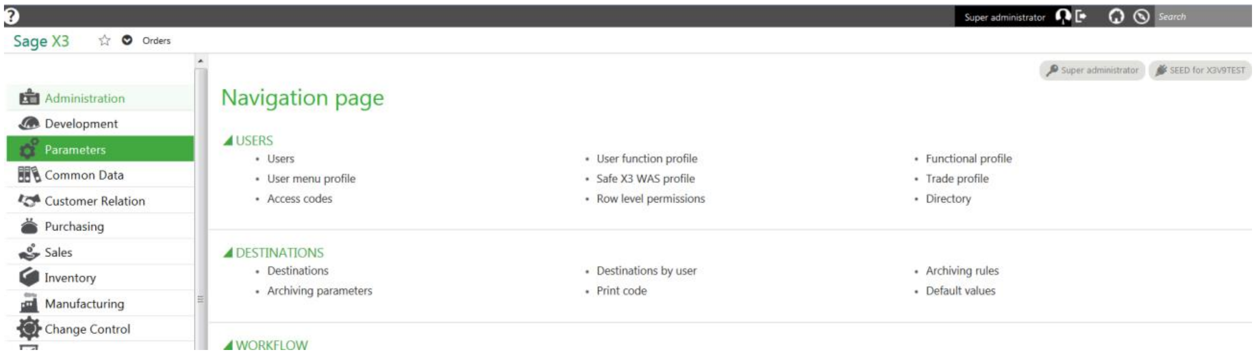
This document has been created using a Sage X3 PU9 patch 6 instance, but applies to any version of Sage X3 from V7 onwards

1. Setup SQL Server tracing using Extended Events from within SQL Server Management Studio for an X3 user session

1.1 Login to X3

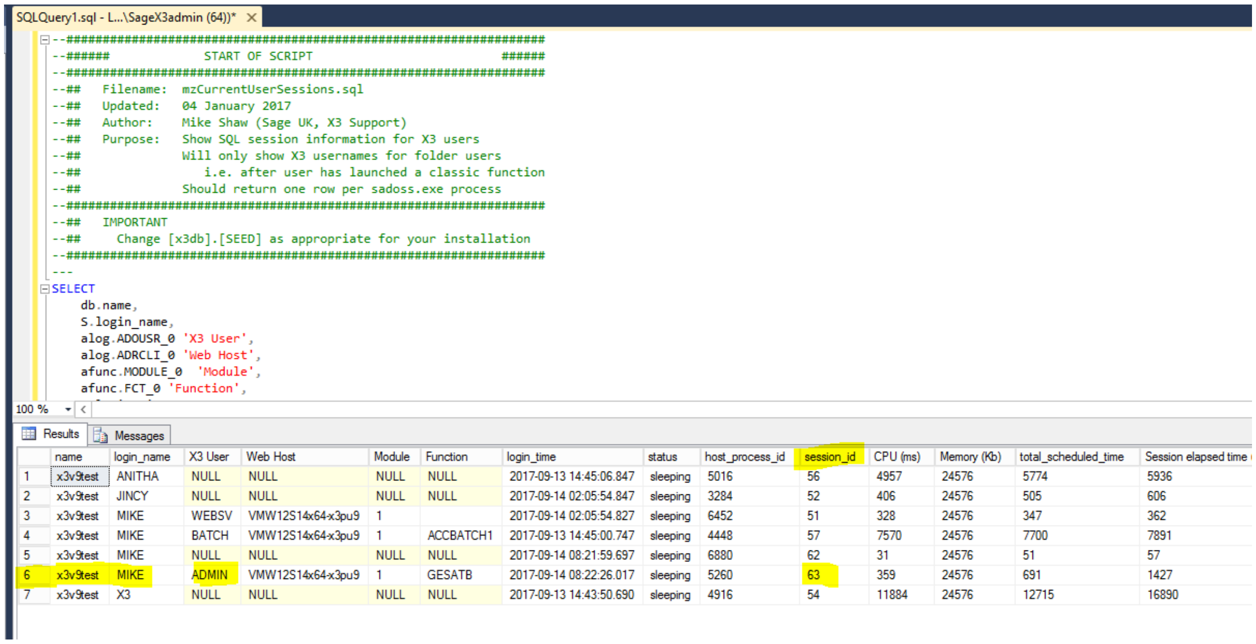
The tracing can only be enabled for a user once they have logged into X3 and launched any Classic function.   This mirrors the way SQL tracing is enabled from within X3 and is because the user session information is only stored in the SQL Server tables once a function has been accessed

a. Login to X3 and select the appropriate folder
b. Launch any classic function, which will initialize a database session for the user
c. Close the function and then get the user to the point at which you want to enable SQL tracing



1.2 Launch SQL Server Management Studio

a. Login to SQL Server Management Studio in the normal way
b. Load the “mzCurrentUserSessions.sql” script (listed in Appendix A of this document)
c. Modify the script, by changing reference to “[$(mzschema)]" to reflect your own folder name
d. Execute the script and review the data returned.
e. Identify your X3 user from the username shown in the script output and note the session_id



```
--######################################################################
--######              START OF SCRIPT                           ######
--######################################################################
--##   Filename:  mzCurrentUserSessions.sql
--##   Updated:   04 January 2017
--##   Author:    Mike Shaw (Sage UK, X3 Support)
--##   Purpose:   Show SQL session information for X3 users
--##              Will only show X3 usernames for folder users
--##                 i.e. after user has launched a classic function
--##              Should return one row per sadoss.exe process
--######################################################################
--##   IMPORTANT
--##     Change [x3db].[SEED] as appropriate for your installation
--######################################################################
---
SELECT
    db.name,
    S.login_name,
    alog.ADOUSR_0 'X3 User',
    alog.ADRCLI_0 'Web Host',
    afunc.MODULE_0  'Module',
    afunc.FCT_0 'Function',
```

	name	login_name	X3 User	Web Host	Module	Function	login_time	status	host_process_id	session_id	CPU (ms)	Memory (Kb)	total_scheduled_time	Session elapsed time
1	x3v9test	ANITHA	NULL	NULL	NULL	NULL	2017-09-13 14:45:06.847	sleeping	5016	56	4957	24576	5774	5936
2	x3v9test	JINCY	NULL	NULL	NULL	NULL	2017-09-14 02:05:54.847	sleeping	3284	52	406	24576	505	606
3	x3v9test	MIKE	WEBSV	VMW12S14x64-x3pu9	1		2017-09-14 02:05:54.827	sleeping	6452	51	328	24576	347	362
4	x3v9test	MIKE	BATCH	VMW12S14x64-x3pu9	1	ACCBATCH1	2017-09-13 14:45:00.747	sleeping	4448	57	7570	24576	7700	7891
5	x3v9test	MIKE	NULL	NULL	NULL	NULL	2017-09-14 08:21:59.697	sleeping	6880	62	31	24576	51	57
6	x3v9test	MIKE	ADMIN	VMW12S14x64-x3pu9	1	GESATB	2017-09-14 08:22:26.017	sleeping	5260	63	359	24576	691	1427
7	x3v9test	X3	NULL	NULL	NULL	NULL	2017-09-13 14:43:50.690	sleeping	4916	54	11884	24576	12715	16890

Another alternative is to login to X3 as a different user, then:

- Navigate to Development, Utilities, Verifications, System monitor, User Monitoring (APSADX)
- Locate the row of the user whose actions you want to trace and click on the ID 1 link or the Actions card and then select Display
- In the Active Processes block, locate the row with “sadoss” in the Processes column and make note of the Process number here

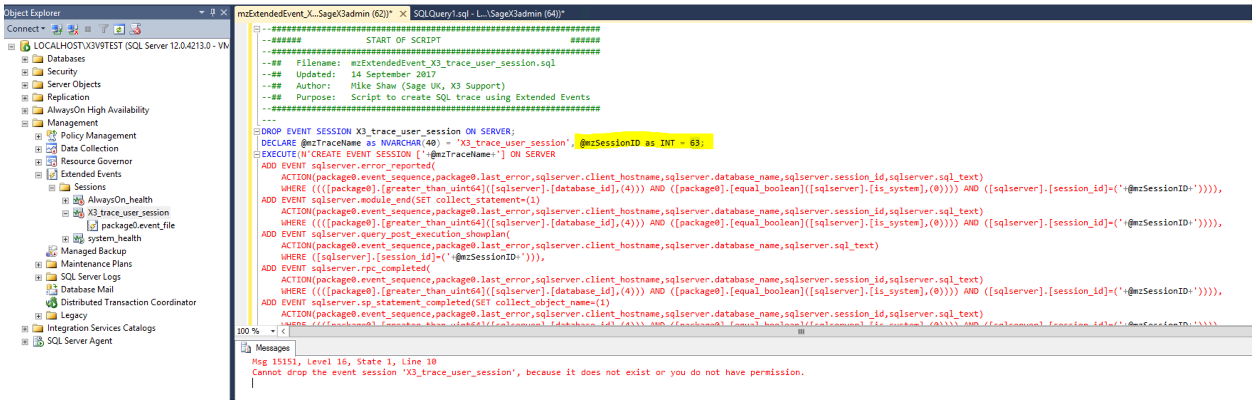
1.3 Create new Extended Event

Load the SQL file “mzExtendedEvent_X3_trace_user_session.sql” (also listed in Appendix B of this document)

Modify the value of the variable “`mzSessionID`” to that of the session_id determined in the previous step

Execute the script to create the Extended Event Session

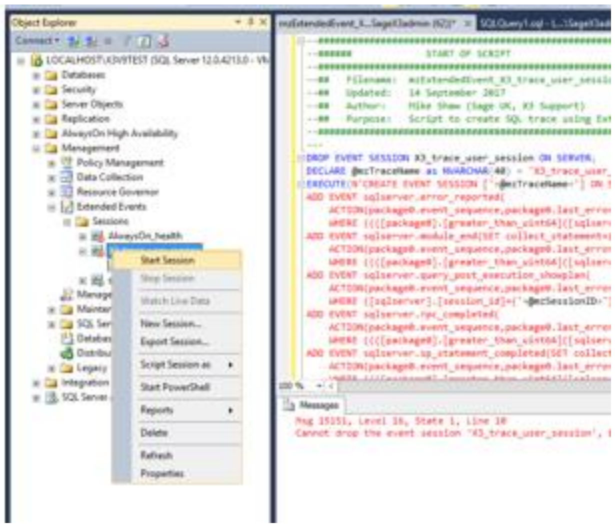
NOTE: the first time you run this script you will see an error “`Cannot drop the event session 'X3_trace_user_session', because it does not exist or you do not have permission`”



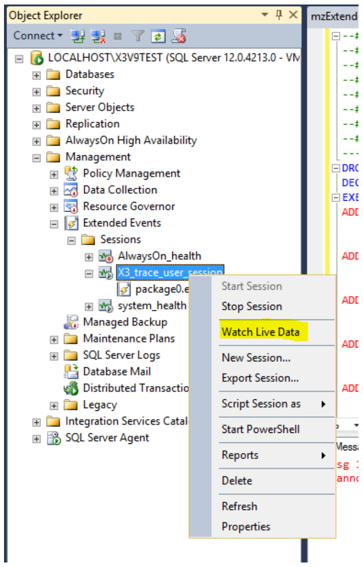
```
--##################################################################
--######              START OF SCRIPT                         ######
--##################################################################
--##   Filename:  mzExtendedEvent_X3_trace_user_session.sql
--##   Updated:   14 September 2017
--##   Author:    Mike Shaw (Sage UK, X3 Support)
--##   Purpose:   Script to create SQL trace using Extended Events
--##################################################################
---
DROP EVENT SESSION X3_trace_user_session ON SERVER;
DECLARE @mzTraceName as NVARCHAR(40) = 'X3_trace_user_session', @mzSessionID as INT = 63;
EXECUTE(N'CREATE EVENT SESSION ['+@mzTraceName+'] ON SERVER
ADD EVENT sqlserver.error_reported(
    ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.database_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE ((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND ([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND ([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.module_end(SET collect_statement=(1)
    ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.database_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE ((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND ([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND ([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.query_post_execution_showplan(
    ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.database_name,sqlserver.sql_text)
    WHERE ([sqlserver].[session_id]=('+@mzSessionID+'))),
ADD EVENT sqlserver.rpc_completed(
    ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.database_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE ((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND ([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND ([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.sp_statement_completed(SET collect_object_name=(1)
    ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.database_name,sqlserver.session_id,sqlserver.sql_text)
```

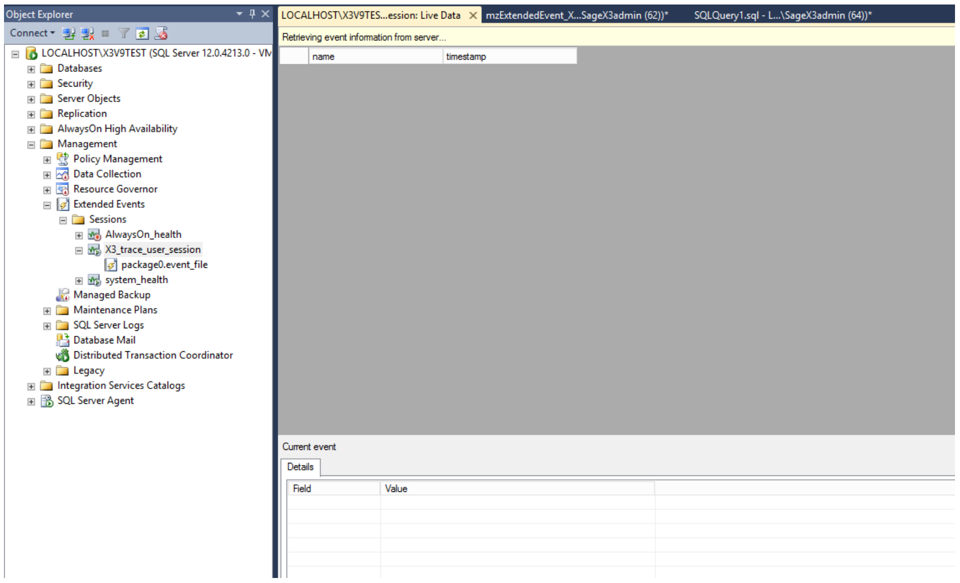
```
Msg 15151, Level 16, State 1, Line 10
Cannot drop the event session 'X3_trace_user_session', because it does not exist or you do not have permission.
```

Start the tracing session by right clicking “x3_trace_user_session” and selecting “Start Session”   You will see the red cross icon turn to a green arrow



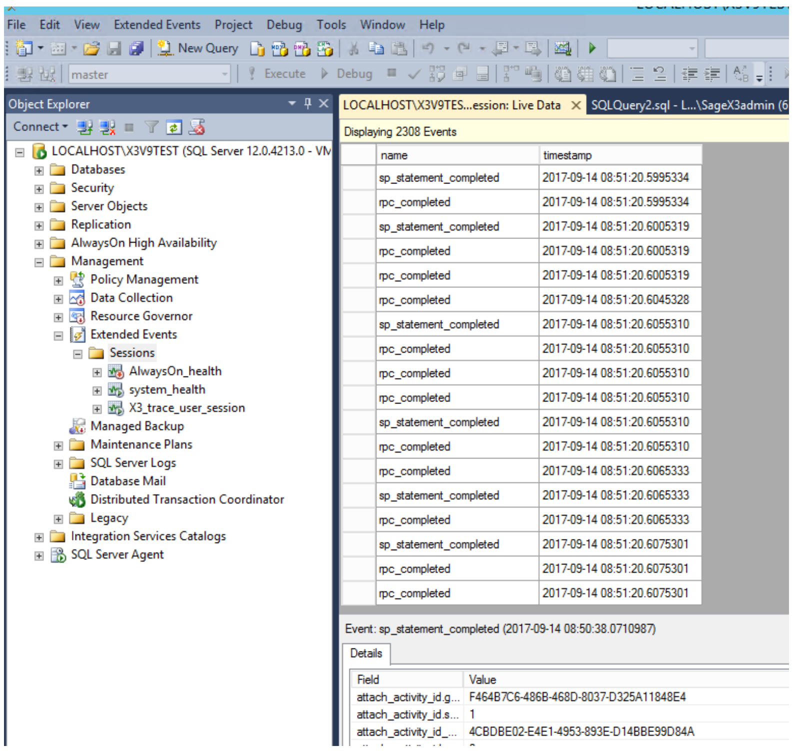
Right click again and select “Watch Live data” which will allow you to check the data is being gathered correctly.   At this stage, nothing should appear in the list of data. If it does, you may have picked the wrong Session ID number, or the user has already started performing some steps


Object Explorer
Connect
LOCALHOST\X3V9TEST (SQL Server 12.0.4213.0 - VM
Databases
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Policy Management
Data Collection
Resource Governor
Extended Events
Sessions
AlwaysOn_health
X3_trace_user_session
package0.e
system_health
Managed Backup
Maintenance Plans
SQL Server Logs
Database Mail
Distributed Transactio
Legacy
Integration Services Catal
SQL Server Agent
Start Session
Stop Session
Watch Live Data
New Session...
Export Session...
Script Session as
Start PowerShell
Reports
Delete
Refresh
Properties


Object Explorer
Connect
LOCALHOST\X3V9TEST (SQL Server 12.0.4213.0 - VM
Databases
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Policy Management
Data Collection
Resource Governor
Extended Events
Sessions
AlwaysOn_health
X3_trace_user_session
package0.event_file
system_health
Managed Backup
Maintenance Plans
SQL Server Logs
Database Mail
Distributed Transaction Coordinator
Legacy
Integration Services Catalogs
SQL Server Agent
LOCALHOST\X3V9TES...ession: Live Data
mzExtendedEvent_X...SageX3admin (62))*
SQLQuery1.sql - L...\SageX3admin (64))*
Retrieving event information from server...
name
timestamp
Current event
Details
Field
Value

1.4 Run the steps in the X3 session that need to be traced

Back in the X3 user session, perform whatever steps which you need to trace. Whilst the user is performing some actions, you will see the "Live Data" information being accumulated



Displaying 2308 Events

name	timestamp
sp_statement_completed	2017-09-14 08:51:20.5995334
rpc_completed	2017-09-14 08:51:20.5995334
sp_statement_completed	2017-09-14 08:51:20.6005319
rpc_completed	2017-09-14 08:51:20.6005319
rpc_completed	2017-09-14 08:51:20.6005319
rpc_completed	2017-09-14 08:51:20.6045328
sp_statement_completed	2017-09-14 08:51:20.6055310
rpc_completed	2017-09-14 08:51:20.6055310
rpc_completed	2017-09-14 08:51:20.6055310
rpc_completed	2017-09-14 08:51:20.6055310
sp_statement_completed	2017-09-14 08:51:20.6055310
rpc_completed	2017-09-14 08:51:20.6055310
rpc_completed	2017-09-14 08:51:20.6065333
sp_statement_completed	2017-09-14 08:51:20.6065333
rpc_completed	2017-09-14 08:51:20.6065333
sp_statement_completed	2017-09-14 08:51:20.6075301
rpc_completed	2017-09-14 08:51:20.6075301
rpc_completed	2017-09-14 08:51:20.6075301

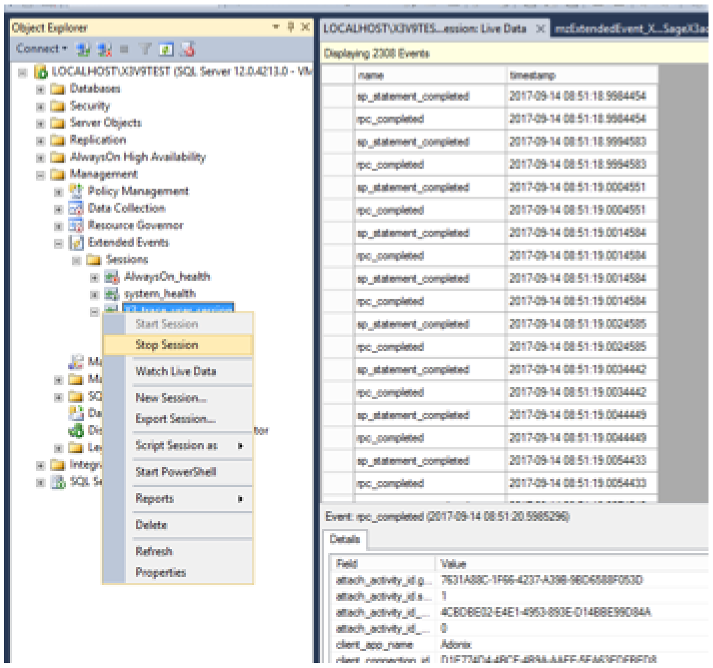
Event: sp_statement_completed (2017-09-14 08:50:38.0710987)

Details

Field	Value
attach_activity_id.g...	F464B7C6-486B-468D-8037-D325A11848E4
attach_activity_id.s...	1
attach_activity_id_...	4CBDBE02-E4E1-4953-893E-D14BBE99D84A

1.5 Stop the SQL trace

Once the user activity has been completed, right click the "x3_trace_user_session" and select "Stop Session" You will see the green arrow icon turn to a red cross



Displaying 2308 Events

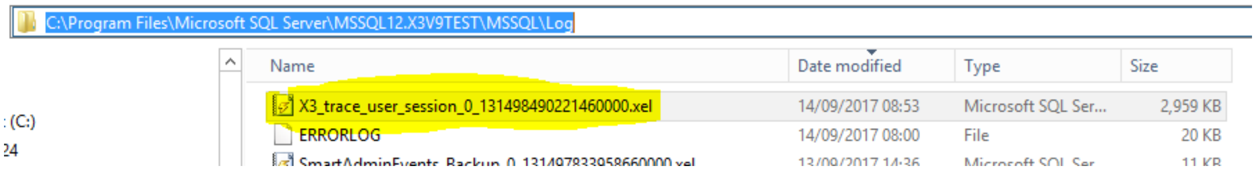
name	timestamp
sp_statement_completed	2017-09-14 08:51:18.9984454
rpc_completed	2017-09-14 08:51:18.9984454
sp_statement_completed	2017-09-14 08:51:18.9994583
rpc_completed	2017-09-14 08:51:18.9994583
sp_statement_completed	2017-09-14 08:51:19.0004551
rpc_completed	2017-09-14 08:51:19.0004551
sp_statement_completed	2017-09-14 08:51:19.0014584
rpc_completed	2017-09-14 08:51:19.0014584
sp_statement_completed	2017-09-14 08:51:19.0014584
rpc_completed	2017-09-14 08:51:19.0014584
sp_statement_completed	2017-09-14 08:51:19.0024585
rpc_completed	2017-09-14 08:51:19.0024585
sp_statement_completed	2017-09-14 08:51:19.0034442
rpc_completed	2017-09-14 08:51:19.0034442
sp_statement_completed	2017-09-14 08:51:19.0044449
rpc_completed	2017-09-14 08:51:19.0044449
sp_statement_completed	2017-09-14 08:51:19.0054433
rpc_completed	2017-09-14 08:51:19.0054433

Event: rpc_completed (2017-09-14 08:51:20.5985296)

Details

Field	Value
attach_activity_id.g...	7631A88C-1F66-4237-A398-9BD6588F053D
attach_activity_id.s...	1
attach_activity_id_...	4CBDBE02-E4E1-4953-893E-D14BBE99D84A
attach_activity_id_...	0
client_app_name	Adonix
client_connection_id	D1E774D4-4BCF-4B9A-AAFF-5EA63FDFBFD8

Locate and upload to Sage Support the ".XEL" file created for this session. This file will be located in the SQL Server log directory, for example "C:\Program Files\Microsoft SQL Server\MSSQL12.X3V9TEST\MSSQL\Log"



C:\Program Files\Microsoft SQL Server\MSSQL12.X3V9TEST\MSSQL\Log

Name	Date modified	Type	Size
X3_trace_user_session_0_131498490221460000.xel	14/09/2017 08:53	Microsoft SQL Ser...	2,959 KB
ERRORLOG	14/09/2017 08:00	File	20 KB
SmartAdminEvents_Backup_0_131497833958660000.xel	13/09/2017 14:36	Microsoft SQL Ser...	11 KB

NOTE : If you need to trace the session for a user after the user logs out of X3, or if they create a new X3 session by using a new browser tab, then you will need to repeat the above steps from step 1.2 onwards to identify the Session ID, and then re-create the Extended Event Session for the new Session ID

Conclusion

This document provides the process that can be used to generate a SQL trace for a specific X3 user's activity, where the SQL trace option cannot be enabled through the Sage X3 user interface

Appendix A - mzCurrentUserSessions.sql

```
--##################################################################
--######              START OF SCRIPT                        ######
--##################################################################
--##   Filename:  mzCurrentUserSessions.sql
--##   Updated:   22 March 2022
--##   Author:    Mike Shaw (Sage UK, X3 Support)
--##   Purpose:   Show SQL session information for X3 users
--##              Will only show X3 usernames for folder users
--##                 i.e. after user has launched a classic function
--##              Should return one row per sadoss.exe process
--##################################################################
--##   IMPORTANT
--##     Change [$(mzschema)] as appropriate for your folder name
--##################################################################
---
SELECT
       S.session_id as "Session Id",
       db.name as "DB Name",
       S.program_name as "Program",
       S.login_name as "Login Name",
       alog.ADOUSR_0 as "EM User",
       alog.ADRCLI_0 as "Web Host",
       afunc.MODULE_0 as "Module",
       afunc.FCT_0 as "Function",
       FORMAT(S.login_time,'dd/MM/yyyy HH:mm:ss') as "Login",
       S.status as "Status",
       S.host_process_id as "Process Id",
       S.cpu_time/1000 as "CPU (secs)",
       S.memory_usage*8192 as "Memory (Kb)",
       S.total_scheduled_time/1000 as "Scheduled time (secs)",
       S.total_elapsed_time/1000 as "Session elapsed time (secs)",
       FORMAT(S.last_request_start_time,'dd/MM/yyyy HH:mm:ss') as "Last Request Start
Time",
       FORMAT(S.last_request_end_time,'dd/MM/yyyy HH:mm:ss') as "Last Request End Time",
       S.reads as "Session Reads",
       S.writes as "Session Writes",
       S.logical_reads as "Session Logical Reads",
       p.blocked as "Blocking Session Id",
       p.waittime as "Wait (ms)"
FROM sys.dm_exec_sessions AS S WITH (NOLOCK)
       INNER JOIN sys.sysprocesses AS p WITH (NOLOCK)
              ON S.session_id = p.spid
       LEFT OUTER JOIN [$(mzschema)].[ALOGIN] AS alog WITH (NOLOCK)
              ON S.session_id = alog.BDDID_0
              and FLG_0 = 2
       LEFT OUTER JOIN [X3].[AFCTCUR] as afunc WITH (NOLOCK)
              ON alog.ADOID_0 = afunc.UID_0
       INNER JOIN sys.databases AS db
              ON S.database_id = db.database_id
WHERE S.is_user_process = 1
       and db.name != 'master'
       and S.program_name in ('Adonix','sadoss.exe')
ORDER BY db.name, S.program_name, S.login_name;
---
--##################################################################
--######                END OF SCRIPT                        ######
--##################################################################
```

If you have a problem running the above SQL, a simplified version is below:

```
select BDDID_0
from <FOLDER>.ALOGIN
where ADOLOG_0 = '<X3userName>'
order by DATCNX_0 desc
```

Where :

`<FOLDER>` is the X3 folder the user is connected to

`<X3userName>` is the X3 user name

Appendix B – mzExtendedEvent_X3_trace_user_session.sql

```
--####################################################################
--######              START OF SCRIPT                          ######
--####################################################################
--##   Filename:  mzExtendedEvent_X3_trace_user_session.sql
--##   Updated:   22 March 2022
--##   Author:    Mike Shaw (Sage UK, X3 Support)
--##   Purpose:   Script to create SQL trace using Extended Events
--####################################################################
--##   IMPORTANT
--##     Change "@mzSessionID as INT = 63" for your sessionId
--####################################################################
---
DROP EVENT SESSION X3_trace_user_session ON SERVER;
DECLARE @mzTraceName as NVARCHAR(40) = 'X3_trace_user_session', @mzSessionID as INT = 63;
EXECUTE(N'CREATE EVENT SESSION ['+@mzTraceName+'] ON SERVER
ADD EVENT sqlserver.error_reported(

ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.da
tabase_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE (((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND
([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND
([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.module_end(SET collect_statement=(1)

ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.da
tabase_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE (((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND
([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND
([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.query_post_execution_showplan(

ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.da
tabase_name,sqlserver.sql_text)
    WHERE ([sqlserver].[session_id]=('+@mzSessionID+'))),
ADD EVENT sqlserver.rpc_completed(

ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.da
tabase_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE (((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND
([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND
([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.sp_statement_completed(SET collect_object_name=(1)

ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.da
tabase_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE (((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND
([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND
([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.sql_batch_completed(

ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.da
tabase_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE (((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND
([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND
([sqlserver].[session_id]=('+@mzSessionID+')))),
ADD EVENT sqlserver.sql_statement_completed(

ACTION(package0.event_sequence,package0.last_error,sqlserver.client_hostname,sqlserver.da
tabase_name,sqlserver.session_id,sqlserver.sql_text)
    WHERE (((([package0].[greater_than_uint64]([sqlserver].[database_id],(4))) AND
([package0].[equal_boolean]([sqlserver].[is_system],(0)))) AND
([sqlserver].[session_id]=('+@mzSessionID+'))))
ADD TARGET package0.event_file(SET
filename=N'''+@mzTraceName+''',max_file_size=(2048),max_rollover_files=(5))
WITH (MAX_MEMORY=4096
KB,EVENT_RETENTION_MODE=ALLOW_SINGLE_EVENT_LOSS,MAX_DISPATCH_LATENCY=30
SECONDS,MAX_EVENT_SIZE=0
KB,MEMORY_PARTITION_MODE=NONE,TRACK_CAUSALITY=ON,STARTUP_STATE=OFF)
');
```

```
---
--####################################################################
--######                END OF SCRIPT                          ######
--####################################################################
```