Deadlock graph using WMI(Windows Management Instrumentaion):

WMI event alert:-

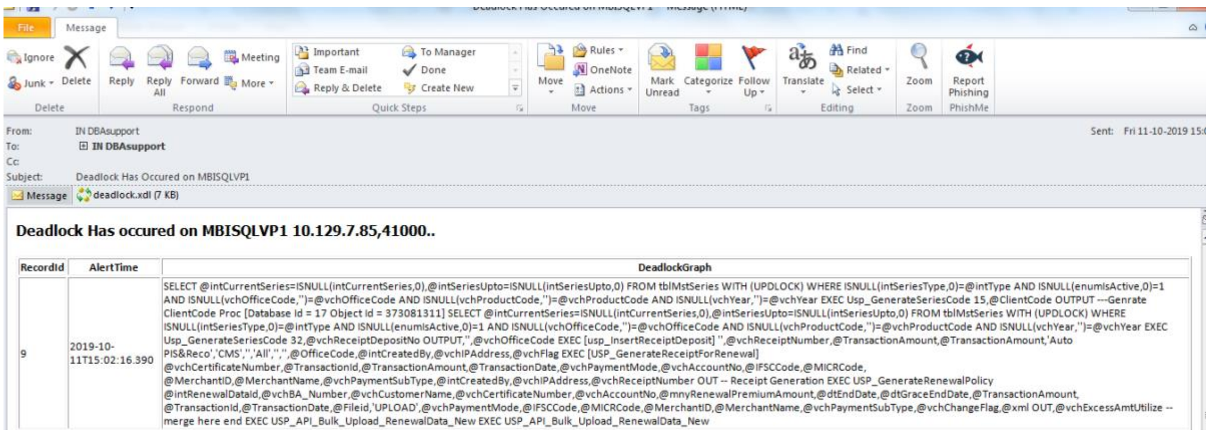
This fires an event in response to a Windows Management Instrumentation (WMI) event, such as when a new file appears in a specific folder.

Advantages:

1. It is in readable format.
2. The graph actually attached with mail.
3. No unnecessary mail triggered.
4. Deadlock information is live scenario base.
5. Previous deadlock entries getting stored in dba database.
6. It detects deadlock information from default trace, no trace flag 1204 and 1222 is not required.

Preview:

Email:



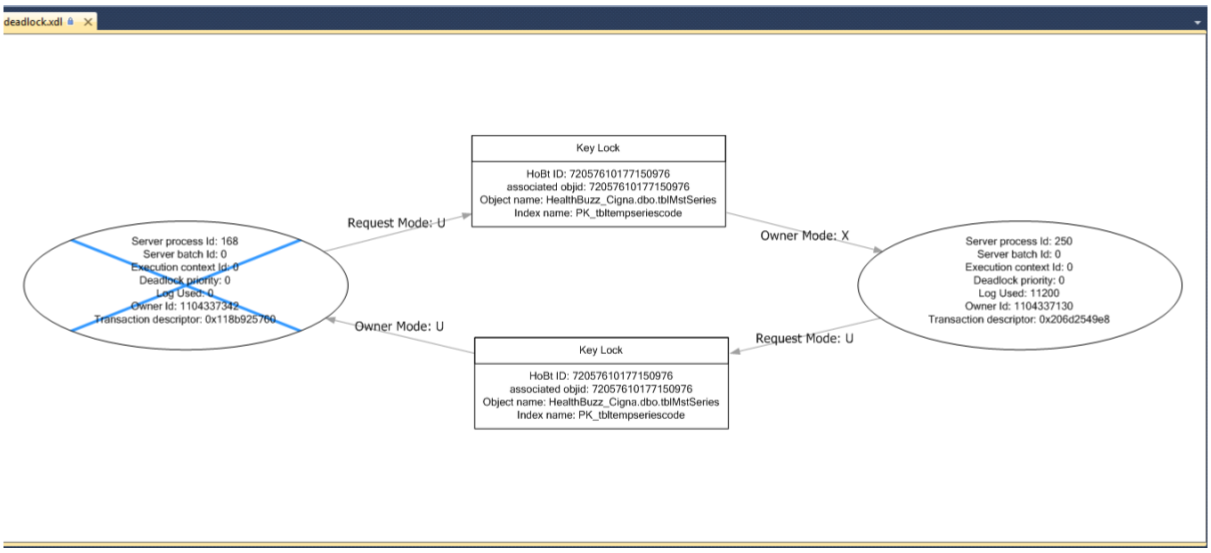
From: IN DBAsupport
To: IN DBAsupport
Cc:
Subject: Deadlock Has Occured on MBISQLVP1
Sent: Fri 11-10-2019 15:

Message deadlock.xdl (7 KB)

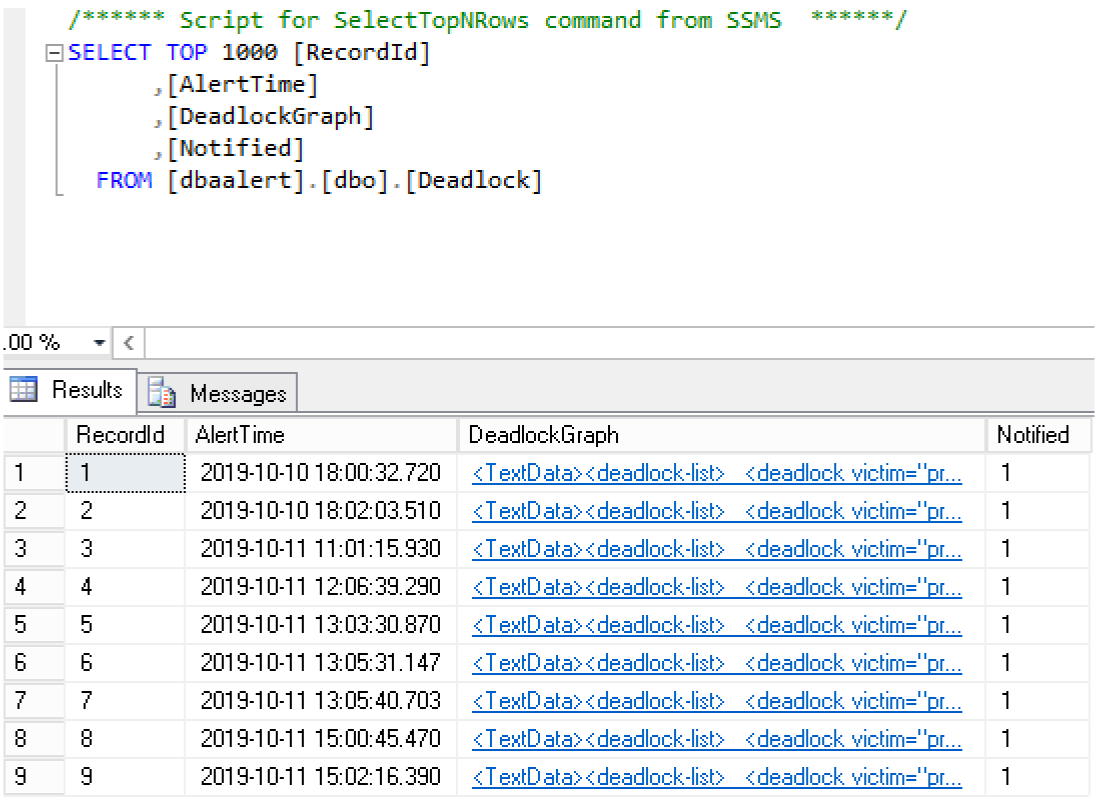
Deadlock Has occured on MBISQLVP1 10.129.7.85,41000..

RecordId	AlertTime	DeadlockGraph
9	2019-10-11T15:02:16.390	SELECT @intCurrentSeries=ISNULL(intCurrentSeries,0),@intSeriesUpto=ISNULL(intSeriesUpto,0) FROM tblMstSeries WITH (UPDLOCK) WHERE ISNULL(intSeriesType,0)=@intType AND ISNULL(enumIsActive,0)=1 AND ISNULL(vchOfficeCode,'')=@vchOfficeCode AND ISNULL(vchProductCode,'')=@vchProductCode AND ISNULL(vchYear,'')=@vchYear EXEC Usp_GenerateSeriesCode 15,@ClientCode OUTPUT ---Genrate ClientCode Proc [Database Id = 17 Object Id = 373081311] SELECT @intCurrentSeries=ISNULL(intCurrentSeries,0),@intSeriesUpto=ISNULL(intSeriesUpto,0) FROM tblMstSeries WITH (UPDLOCK) WHERE ISNULL(intSeriesType,0)=@intType AND ISNULL(enumIsActive,0)=1 AND ISNULL(vchOfficeCode,'')=@vchOfficeCode AND ISNULL(vchProductCode,'')=@vchProductCode AND ISNULL(vchYear,'')=@vchYear EXEC Usp_GenerateSeriesCode 32,@vchReceiptDepositNo OUTPUT,'',@vchOfficeCode EXEC [usp_InsertReceiptDeposit] '',@vchReceiptNumber,@TransactionAmount,@TransactionAmount,'Auto PIS&Reco','CMS','','All','','',@OfficeCode,@intCreatedBy,@vchIPAddress,@vchFlag EXEC [USP_GenerateReceiptForRenewal] @vchCertificateNumber,@TransactionId,@TransactionAmount,@TransactionDate,@vchPaymentMode,@vchAccountNo,@IFSCCode,@MICRCode, @MerchantID,@MerchantName,@vchPaymentSubType,@intCreatedBy,@vchIPAddress,@vchReceiptNumber OUT -- Receipt Generation EXEC USP_GenerateRenewalPolicy @intRenewalDataId,@vchBA_Number,@vchCustomerName,@vchCertificateNumber,@vchAccountNo,@mnyRenewalPremiumAmount,@dtEndDate,@dtGraceEndDate,@TransactionAmount, @TransactionId,@TransactionDate,@FileId,'UPLOAD',@vchPaymentMode,@IFSCCode,@MICRCode,@MerchantID,@MerchantName,@vchPaymentSubType,@vchChangeFlag,@xml OUT,@vchExcessAmtUtilize -- merge here end EXEC USP_API_Bulk_Upload_RenewalData_New EXEC USP_API_Bulk_Upload_RenewalData_New

Graph:


deadlock.xdl
Key Lock
HoBt ID: 72057610177150976
associated objid: 72057610177150976
Object name: HealthBuzz_Cigna.dbo.tblMstSeries
Index name: PK_tbltempseriescode
Request Mode: U
Owner Mode: X
Server process Id: 168
Server batch Id: 0
Execution context Id: 0
Deadlock priority: 0
Log Used: 0
Owner Id: 1104337342
Transaction descriptor: 0x118b925760
Server process Id: 250
Server batch Id: 0
Execution context Id: 0
Deadlock priority: 0
Log Used: 11200
Owner Id: 1104337130
Transaction descriptor: 0x206d2549e8
Owner Mode: U
Request Mode: U
Key Lock
HoBt ID: 72057610177150976
associated objid: 72057610177150976
Object name: HealthBuzz_Cigna.dbo.tblMstSeries
Index name: PK_tbltempseriescode

Deadlock entries in table:



```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [RecordId]
      ,[AlertTime]
      ,[DeadlockGraph]
      ,[Notified]
  FROM [dbaalert].[dbo].[Deadlock]
```

	RecordId	AlertTime	DeadlockGraph	Notified
1	1	2019-10-10 18:00:32.720	<TextData><deadlock-list> <deadlock victim="pr...	1
2	2	2019-10-10 18:02:03.510	<TextData><deadlock-list> <deadlock victim="pr...	1
3	3	2019-10-11 11:01:15.930	<TextData><deadlock-list> <deadlock victim="pr...	1
4	4	2019-10-11 12:06:39.290	<TextData><deadlock-list> <deadlock victim="pr...	1
5	5	2019-10-11 13:03:30.870	<TextData><deadlock-list> <deadlock victim="pr...	1
6	6	2019-10-11 13:05:31.147	<TextData><deadlock-list> <deadlock victim="pr...	1
7	7	2019-10-11 13:05:40.703	<TextData><deadlock-list> <deadlock victim="pr...	1
8	8	2019-10-11 15:00:45.470	<TextData><deadlock-list> <deadlock victim="pr...	1
9	9	2019-10-11 15:02:16.390	<TextData><deadlock-list> <deadlock victim="pr...	1

Configuration:

```sql
/*Requires sysadmin permission to create and execute this job*/

create database dbaalert
go
```

```
use dbaalert
go
IF EXISTS (SELECT * FROM dbo.sysobjects WHERE id = OBJECT_ID(N'[dbo].[Deadlock]') AND
OBJECTPROPERTY(id, N'IsUserTable') = 1)
Print ' *** IF Table Deadlock Table Already Exists... moving Further *** '
ELSE
create table Deadlock
(
RecordId int identity (1,1) primary key not null,
AlertTime datetime not null,
DeadlockGraph xml,
Notified int not null constraint [DF_deadlock_flag] default (0)
)
go
create index deadlock_idx on Deadlock  (AlertTime) with fillfactor = 100
go
USE [msdb]
GO
-- No need to enable deadlock trace flags
--dbcc traceon (1204,1222,-1)
--go
-- enable replace runtime tokens for sql agent to respond to WMI alets
USE [msdb]
GO
EXEC msdb.dbo.sp_set_sqlagent_properties @alert_replace_runtime_tokens=1
GO
-- create the job
USE [msdb]
GO


BEGIN TRANSACTION
DECLARE @ReturnCode INT
SELECT @ReturnCode = 0

IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'[Uncategorized
(Local)]' AND category_class=1)
BEGIN
EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB', @type=N'LOCAL',
@name=N'[Uncategorized (Local)]'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)
EXEC @ReturnCode =  msdb.dbo.sp_add_job @job_name=N'DBA Group - monitoring - Deadlock
detector',
              @enabled=1,
              @notify_level_eventlog=0,
              @notify_level_email=0,
              @notify_level_netsend=0,
              @notify_level_page=0,
              @delete_level=0,
              @description=N'
Funtionality: 1. Real time Deadlock detection using WMI
2. Send email out when deadlock occurs along with
Deadlock Graph attached.
3. Logs all the deadlocks in a central Table
dbamaint.dbo.Deadlock
4. Does not require trace flags 1204 and 1222 enabled',
              @category_name=N'[Uncategorized (Local)]',
              @owner_login_name=N'sa', @job_id = @jobId OUTPUT
```

```
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback


EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'Insert
Deadlock info',
              @step_id=1,
              @cmdexec_success_code=0,
              @on_success_action=3,
              @on_success_step_id=0,
              @on_fail_action=2,
              @on_fail_step_id=0,
              @retry_attempts=0,
              @retry_interval=0,
              @os_run_priority=0, @subsystem=N'TSQL',
              @command=N'INSERT INTO Deadlock (

AlertTime,

DeadlockGraph

)

VALUES (

GETDATE(),

''$(ESCAPE_NONE(WMI(TextData)))''

)',
              @database_name=N'dbaalert',
              @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback


EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'Send Email
with Deadlock Graph',
              @step_id=2,
              @cmdexec_success_code=0,
              @on_success_action=1,
              @on_success_step_id=0,
              @on_fail_action=2,
              @on_fail_step_id=0,
              @retry_attempts=0,
              @retry_interval=0,
              @os_run_priority=0, @subsystem=N'TSQL',
              @command=N'if exists (select 1 from dbo.Deadlock where notified = 0 )
begin
declare @tableHTML nvarchar(max)
set @tableHTML =N''<H3><FONT SIZE="3" FACE="Tahoma">Deadlock Has occured on
''+@@servername+'' ..  </FONT></H3>''
set @tableHTML = @tableHTML+ N''<table border="1">'' +
           N''<FONT SIZE="2" FACE="Calibri">'' +
            N''<tr><th align="center">RecordId</th>'' +
                     N''<th align="center">AlertTime</th>'' +
            N''<th align="center">DeadlockGraph</th>'' +

                     N''</tr>'' +
           ISNULL(CAST ( (
                                               select  td = '''',
                                                             td = RecordId,'''',
                                        td = AlertTime,'''',
                                                             td = DeadlockGraph,''''
```

```
                        from dbo.Deadlock where notified = 0

        FOR XML PATH(''tr''), TYPE

             ) AS NVARCHAR(MAX) ),'''') +
             N''</FONT>'' +
             N''</table>'' ;

exec master..xp_cmdshell 'bcp "SELECT  [Deadlockgraph].query(''/TextData/deadlock-
list'') FROM dbaalert..[Deadlock] where Notified = 0" queryout
"c:\deadlock_test\deadlock.xdl" -c -q -T -S MBISQLVP1';

-- send email out with the graph attached
declare @subject1 varchar(50)
set @subject1 = ''Deadlock Has Occured on ''+@@servername
EXEC msdb.dbo.sp_send_dbmail
@profile_name = ''DBASUPPORT_profile'',
            @recipients=''dbasupport@manipalcigna.com'',
            @subject = @subject1,
            @body = @tableHTML,
            @body_format = ''HTML'',
            @file_attachments = ''c:\deadlock\deadlock.xdl'';
end
go
-- update the Deadlock table so that when the job runs it wont send out previous alert
update dbaalert.dbo.Deadlock
set Notified = 1 where notified = 0
--SELECT * FROM dbaalert.dbo.Deadlock',
              @database_name=N'dbaalert',
              @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id = 1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name =
N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
COMMIT TRANSACTION
GOTO EndSave
QuitWithRollback:
    IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:

-- create an WMI alert to respond to deadlocks
IF EXISTS (SELECT name FROM msdb.dbo.sysalerts WHERE name = N'Respond to Deadlock')
EXEC msdb.dbo.sp_delete_alert @name=N'Respond to Deadlock'
GO
DECLARE @server_namespace varchar(255)
IF ISNULL(CHARINDEX('\', @@SERVERNAME), 0) > 0
SET @server_namespace = N'\\.\root\Microsoft\SqlServer\ServerEvents\' +
SUBSTRING(@@SERVERNAME, ISNULL(CHARINDEX('\', @@SERVERNAME), 0) + 1, LEN(@@SERVERNAME)
- ISNULL(CHARINDEX('/', @@SERVERNAME), 0))
ELSE
SET @server_namespace = N'\\.\root\Microsoft\SqlServer\ServerEvents\MSSQLSERVER'
EXEC msdb.dbo.sp_add_alert @name=N'Respond to Deadlock',
@wmi_namespace=@server_namespace,
@wmi_query=N'SELECT * FROM DEADLOCK_GRAPH',
@job_name='DBA Group - monitoring - Deadlock detector' ;
```

Creating and Testing Deadlock:

```
--Two global temp tables with sample data for demo purposes.
CREATE TABLE Employees (
    EmpId INT IDENTITY,
    EmpName VARCHAR(16),
    Phone VARCHAR(16)
)
GO

INSERT INTO Employees (EmpName, Phone)
VALUES ('Martha', '800-555-1212'), ('Jimmy', '619-555-8080')
GO

CREATE TABLE Suppliers(
    SupplierId INT IDENTITY,
    SupplierName VARCHAR(64),
    Fax VARCHAR(16)
)
GO

INSERT INTO Suppliers (SupplierName, Fax)
VALUES ('Acme', '877-555-6060'), ('Rockwell', '800-257-1234')
GO


Session 1                    | Session 2
===========================================================
BEGIN TRAN;                  | BEGIN TRAN;
===========================================================
UPDATE Employees
SET EmpName = 'Mary'
WHERE EmpId = 1
===========================================================
                              | UPDATE Suppliers
                              | SET Fax = N'555-1212'
                              | WHERE SupplierId = 1
===========================================================
UPDATE Suppliers
SET Fax = N'555-1212'
WHERE SupplierId = 1
===========================================================
<blocked>                     | UPDATE Employees
                              | SET Phone = N'555-9999'
                              | WHERE EmpId = 1
===========================================================
                              | <blocked>
===========================================================
```