In this video we will discuss **how to read and analyze sql server deadlock information captured in the error log**, so we can understand what's causing the deadlocks and take appropriate actions to prevent or minimize the occurrence of deadlocks. This is continuation to [Part 80](http://csharp-video-tutorials.blogspot.com/2015/08/logging-deadlocks-in-sql-server.html). Please watch [Part 80](http://csharp-video-tutorials.blogspot.com/2015/08/logging-deadlocks-in-sql-server.html) from [SQL Server tutorial](https://www.youtube.com/playlist?list=PL08903FB7ACA1C2FB) before proceeding.   
  
   
  
**The deadlock information in the error log has three sections**

|  |  |
| --- | --- |
| **Section** | **Description** |
| Deadlock Victim | Contains the ID of the process that was selected as the deadlock victim and killed by SQL Server. |
| Process List | Contains the list of the processes that participated in the deadlock. |
| Resource List | Contains the list of the resources (database objects) owned by the processes involved in the deadlock |

**Process List :** The process list has lot of items. Here are some of them that are particularly useful in understanding what caused the deadlock.

|  |  |
| --- | --- |
| **Node** | **Description** |
| loginname | The loginname associated with the process |
| isolationlevel | What isolation level is used |
| procname | The stored procedure name |
| Inputbuf | The code the process is executing when the deadlock occured |

**Resource List :** Some of the items in the resource list that are particularly useful in understanding what caused the deadlock.

|  |  |
| --- | --- |
| **Node** | **Description** |
| objectname | Fully qualified name of the resource involved in the deadlock |
| owner-list | Contains (owner id) the id of the owning process and the lock mode it has acquired on the resource. lock mode determines how the resource can be accessed by concurrent transactions. S for Shared lock, U for Update lock, X for Exclusive lock etc |
| waiter-list | Contains (waiter id) the id of the process that wants to acquire a lock on the resource and the lock mode it is requesting |

To prevent the deadlock that we have in our case, we need to ensure that database objects (Table A & Table B) are accessed in the same order every time.