ORDER BY and IDENTITY

Source:
http://www.tech−archive.net/Archive/SQL−Server/microsoft.public.sqlserver.programming/2004−12/4647.html

---

From: Questar (*Questar_at_discussions.microsoft.com*)
Date: 12/20/04

Date: Mon, 20 Dec 2004 09:25:06 −0800

It has come to our attention that the expected (and observed) behavior of a specific T−SQL statement is not guaranteed. You have probably encountered the issue before, but I will provide two examples here.

Syntax Example 1:

```
CREATE TABLE #T1 (T1 int, Z1 varchar(20))

INSERT #T1 VALUES (1,'X')
INSERT #T1 VALUES (2,'Y')
INSERT #T1 VALUES (3,'Z')

SELECT T1 AS T2, Z1 AS Z2, IDENTITY(int,1,1) AS I2 INTO #T2 FROM #T1 ORDER
BY T1 DESC

DROP TABLE #T1
DROP TABLE #T2
```

Syntax Example 2:

```
CREATE TABLE #T1 (T1 int, Z1 varchar(20))
CREATE TABLE #T2 (T2 int, Z2 varchar(20), I2 int IDENTITY(1,1))

INSERT #T1 VALUES (1,'X')
INSERT #T1 VALUES (2,'Y')
INSERT #T1 VALUES (3,'Z')

INSERT #T2 (T2,Z2) SELECT T1,Z1 FROM #T1 ORDER BY T1 DESC

DROP TABLE #T1
DROP TABLE #T2
```

According to a SQL Server MVP, the SQL Server 2000 execution plans for these two examples are not guaranteed to have the sort occur before the assignment of IDENTITY values. The MVP said the expected behavior was guaranteed in SQL Server 2005. These two examples produce very similar, but not identical, execution plans. Looking at the statements logically I could understand if

the syntax of Example 1 resulted in the Compute Scalar (IDENTITY) operation happening before the Sort operation, but that behavior would seem very wrong for Example 2. I believe there's a significant amount of T−SQL code being used (including some of ours) that depends upon the syntax of Example 2 resulting in the Sort operation happening before the Compute Scalar (IDENTITY) operation.

Does anybody have any additional thoughts or information on this matter?