

SQL Server Variable Declaration Gotcha

Posted At : December 16, 2009 11:12 AM | Posted By : Jon Hartmann

Related Categories: Microsoft Tools

Part of my new job is to work more with Transact-SQL than I've ever done before, so I've been making stored procedures and batches of SQL commands all over the place. Today I ran into a nice little "gotcha" that had me stumped

The problem was that I was setting declaring an *nvarchar(500)* variable, setting its value to a server path, and only seeing '\ ' as the value in the variable. For example:

```
DECLARE @test nvarchar(500)

SET @test = N'\\server\path\'

SELECT @test AS testValue
```

And the result would be '\'. The problem was that in reality, I'd written:

```
DECLARE @test2 nvarchar

SET @test2 = N'\\server\path\'

SELECT @test2 AS testValue
```

Since SQL Server takes *nvarchar* to mean *nvarchar(1)*, it was ignoring the rest of my server path. It was easy to see the source of the problem once I started looking at the *DECLARE* statement directly, but it was much harder to see when I was looking at it as part of a long list of *DECLARE*s like this:

```
DECLARE @PrimaryServer sysname -- The server on which the primary log shipping database is located
DECLARE @PrimaryDatabase sysname -- The database to use as the primary log shipping database
DECLARE @BackupSourceDirectory nvarchar -- Location that log backups should be saved
DECLARE @SecondaryDatabase sysname -- Name of the secondary database
DECLARE @BackupDestination nvarchar(500) -- Location on the secondary server where logs are saved after moving from the primary server
DECLARE @CopyJobName nvarchar(128) -- Name for the job that handles copying log backups
DECLARE @RestoreJobName nvarchar(128) -- Name for the job that handles restoring log backups
DECLARE @FileRetentionPeriod int -- How long, in minutes, that history are retained
DECLARE @CopyScheduleName nvarchar(128) -- Name for the schedule for log copying
DECLARE @RestoreScheduleName nvarchar(128) -- Name for the schedule for log restoration
DECLARE @CopyFreqType int -- Frequency Type for the copy schedule
DECLARE @CopyFreqInterval int -- Frequency Interval for the copy schedule
DECLARE @CopyFreqSubdayType int -- Frequency Subday Type for the copy schedule
DECLARE @CopyFreqSubdayInterval int -- Frequency SubDay Interval for the copy schedule
DECLARE @RestoreFreqType int -- Frequency Type for the restore schedule
DECLARE @RestoreFreqInterval int -- Frequency Interval for the restore schedule
DECLARE @RestoreFreqSubdayType int -- Frequency Subday Type for the restore schedule
DECLARE @RestoreFreqSubdayInterval int -- Frequency SubDay Interval for the restore schedule
DECLARE @HistoryRetentionPeriod int -- How long, in minutes, that the history will be retained
DECLARE @RestoreThreshold int -- How long, in minutes, before an alert is made if the restore job cannot complete
```

So, to make a long story short, if your variables are acting funny in SQL Server, check your *DECLARE* statement.