**Placing a Mark in SQL Server's Transaction Log for Fall Back**

How many times have you had a programmer come to you and say they want you (the DBA) to restore their database to sometime prior to when they accidentally corrupted it? If you are doing FULL transaction logging you can do a point in time recovery to restore the database to just prior to when the corruption occurred. But in order to do that you need to know exactly when the programmer corrupted the data, which in a lot of cases is not known down to the second.

How many times have you had a programmer come to you and say they accidentally ran an adhoc UPDATE, INSERT, or DELETE statement against their database, and now they want you (the DBA) to restore their database to sometime prior to when they accidentally corrupted their DB?  If you have been a DBA for very long you probably have gotten this kind of request more than once.  If you are doing FULL transaction logging you can do a point in time recovery to restore the database to just prior to when the corruption occurred.  But in order to do that you need to know exactly when the programmer corrupted the data.  Which in a lot of cases is not known down to the second.

When the programmer only generally knows when they corrupted the database the DBA might have to go through a number of restore operations in order to find the exact time when the database was corrupted.  Having to restore the database over and over again until you find the right time to stop the restore can be a trial and error process.  There is a better way!

I suggest that you train programmers to create a marked transaction prior to running any adhoc update process.  By doing this all you need to do is know the name of the transaction in order to recover their database prior to when the programmer ran the adhoc process that corrupted the database.

It is quite simple to create a marked transaction.  To do that all the programmer needs to do is to wrap their adhoc code inside a named transaction, like below:

BEGIN TRANSACTION Start\_Adhoc\_Update

    WITH MARK 'Running my Adhoc update process';

<insert adhoc code here

COMMIT TRANSACTION;

Here I have created a marked transaction named “Start\_Adhoc\_Update”, with a description of “Running my Adhoc update process”.  If I find out later that I corrupted the database I can then run a RESTORE LOG statement where I use the  STOPBEFOREMARK option to restore my database to just prior to the “Start\_Adhoc\_Update” mark in the transaction log.