

## [Learn SQL Server in a Month of Lunches](#)

By Grant B. Fritchey

*There's an old saying in computers, your data is only as good as your last backup. There is a little known corollary to that saying; your backups are only as good as your last restore. In this article based on chapter 7 [Learn SQL Server in a Month of Lunches](#), author Grant Fritchey shows how to use log backups in order to get your database to a particular point in time as part of an emergency.*

To save 35% on your next purchase use Promotional Code **fritchey0735** when you check out at [www.manning.com](http://www.manning.com).

[You may also be interested in...](#)

## *Restoring to a Point in Time*

Imagine a situation where all of the people in an office are working and inputting data into a database. Then, a disk fails and you lose the database. You quickly replace the disk and now the business wants to restore as much data as you can. The last full backup was taken the night before, and you will need to use this. You have log backups running every fifteen minutes, so you can get back to, at worst, fifteen minutes before the event. Now you just need to restore to that point in time.

### ***Using the GUI for point in time recovery***

Assuming you have a database that is in Full Recovery mode and that you have log backups run for that database, you can take advantage of the ability to restore to a specific point in time. To use the GUI to make this happen, just right-click on the database in question and, in the context menu, select Tasks, then Restore, and, finally, "Database..." This will open the Restore Database window that you can see in figure 1.

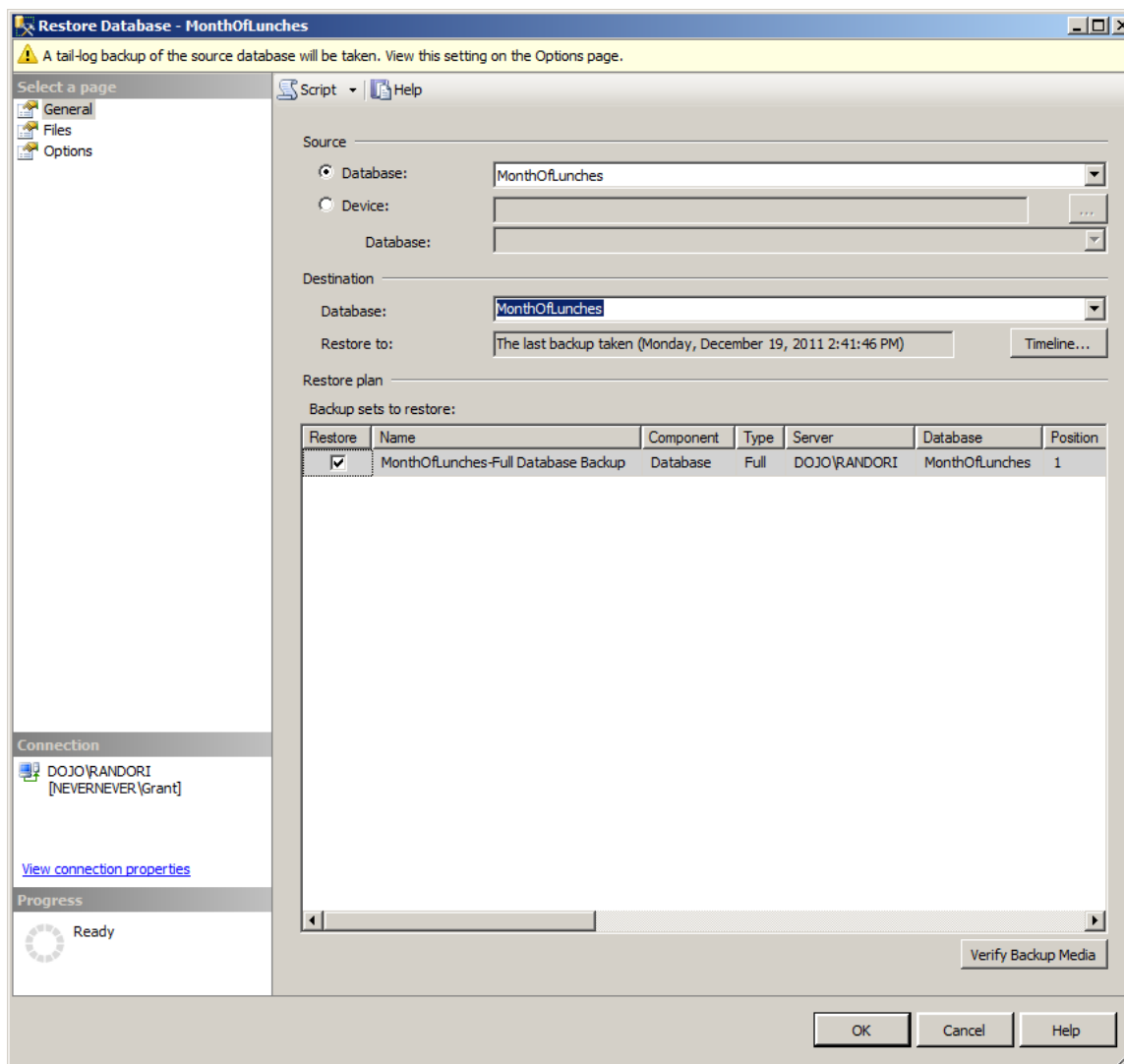


Figure 1 Restore Database window for an existing database

You'll need to be sure that you have both a full backup and a log backup available in the backup sets as shown in figure 2.

Backup sets to restore:						
Restore	Name	Component	Type	Server	Database	Position
<input checked="" type="checkbox"/>	MonthOfLunches-Full Database Backup	Database	Full	DOJO\RANDOMORI	MonthOfLunches	1
<input checked="" type="checkbox"/>	MonthOfLunches_LogBackup_2011-12-19_14-53-32	Log	Transaction Log	DOJO\RANDOMORI	MonthOfLunches	2
<input checked="" type="checkbox"/>	MonthOfLunches_LogBackup_2011-12-19_14-55-22	Log	Transaction Log	DOJO\RANDOMORI	MonthOfLunches	3
<input checked="" type="checkbox"/>	MonthOfLunches_LogBackup_2011-12-24_09-18-44	Log	Transaction Log	DOJO\RANDOMORI	MonthOfLunches	4

Figure 2 Backup sets to restore including the Transaction Log backups needed to restore to a point in time

If you don't have log backups in this list, you'll need to add them. Once you have a full set of backups available, you need to determine the point in time you want to recover to. If you simply want to restore the full database backup and then the log backups, you can run the restore from where you are. But, if you want to adjust the point in time to something particular, you need to click on the Timeline button from the Destination portion of the

For Source Code, Sample Chapters, the Author Forum and other resources, go to  
<http://www.manning.com/fritchey/>

Database Restore window. When you click on the TimeLine button, a new window will open, the Backup Timeline visible in figure 3.

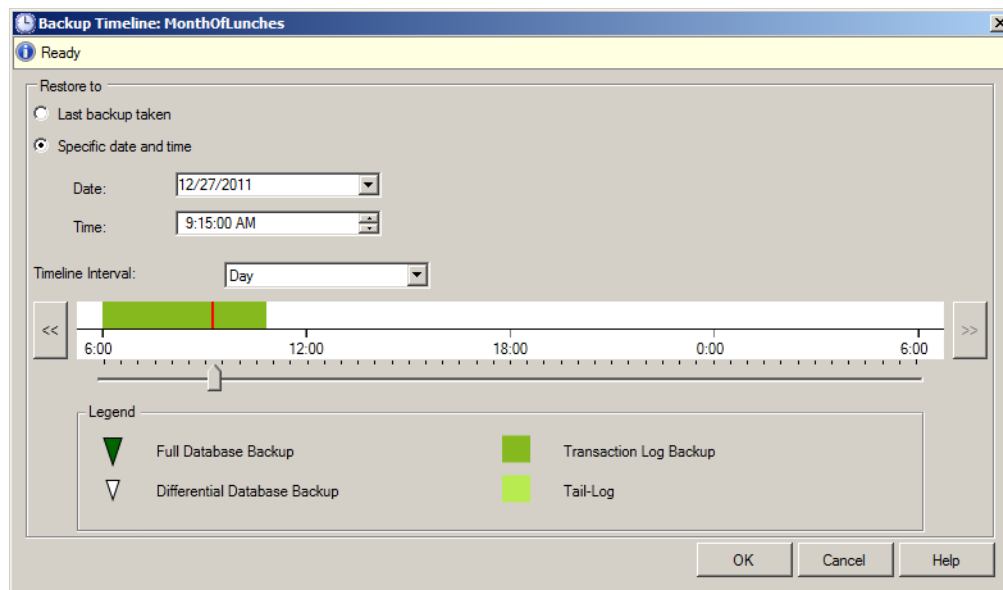


Figure 3 Backup timeline window with a particular point in time selected for restore:

You have two basic options with the window. First, you can use the Last backup taken option, which means you'll restore the latest available backup. If this is a log backup, you'll get the appropriate full backup and then all logs leading up to this log. But that's really no different from running the restore from the previous window.

To see the power of the moment in time restore, you want to select "Specific date and time." Then, all the controls will be enabled. You can use the Date and Time dropdowns to select a moment in time or you can use the sliding bar graph to graphically select a particular date and time. The Time Interval dropdown can be used to change the scale of the sliding bar. For my example, I have the Date set to 12/27/11 and the time set to 9:15:00 AM. Clicking on the OK button will set the date and time to what I've specified. You can see this on the "Restore To" value. I may also see a different list of log backups in the backup sets if the date and time used eliminate the need for other backups. Everything will be displayed for you as you see in figure 4.

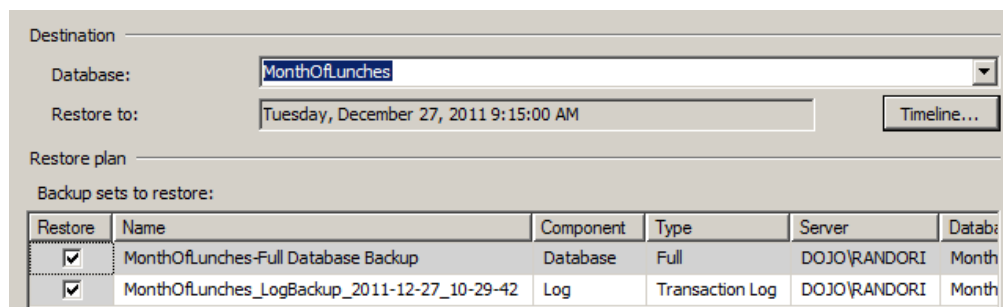


Figure 4 The Restore to and the Backup sets reflect the changes made in the Timeline

You can see that the Restore to: value has changed to show what was selected in the TimeLine. If you now click OK, the restore will run (assuming you've met all the same requirements necessary for a successful restore).

This is one of those places where the GUI is doing a lot for you behind the scene. We'll see exactly what occurred by walking through the same process of restoring to a point in time using T-SQL.

## T-SQL statements for a point in time restore

The basic process is the same. You need to identify the resources needed for the restore operation and you need to tell them to restore to a particular point in time. But, unlike when you used the GUI, you will have to do everything on your own during the restore operation using T-SQL. More is required of you than the simple restore operations.

A major part of the restore operation is a process called recovery. This process makes sure that the restore operation is completed successfully. Because a backup can have committed or uncommitted transactions that took place during the backup process, a restore without taking these into account would leave the database in a broken state. To ensure this is not the case, recovery will commit all completed transactions and roll back all uncompleted transactions so that your database is in the correct state. But, because now you're trying to add log files to our restore operation, you don't want recovery to occur until you get to the particular point in time you're interested in. That changes how the restore process works.

First, you have to restore the full backup as follows:

```
RESTORE DATABASE MonthOfLunches
FROM DISK = N'C:\data\MonthOfLunches_20111116_195228_RANDOM.bak'
WITH NORECOVERY, REPLACE;
```

We're going to RESTORE a database named MonthOfLunches. The source of the backup is provided as FROM a particular file on the DISK. Then, we have to specify that this backup will REPLACE the existing database. We need to specify that we do not want the recovery process to occur. This is done by including the NORECOVERY command in the WITH statement. When this command completes, if you refresh the Object Explorer window, you'll note that your database is not the same as it was. I show my database list in figure 5.

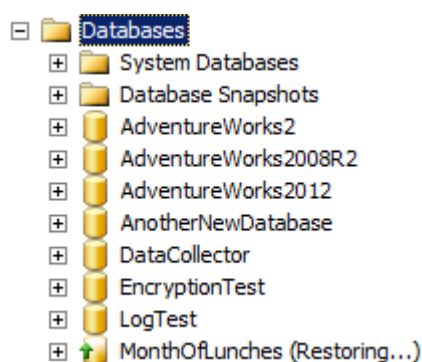


Figure 5 MonthOfLunches database is still restoring

Because I've instructed SQL Server not to finish the restore of the database, you can see that my database is still restoring.

The MonthOfLunches database is now completely offline and waiting for us to finish the process of the restore. I could simply issue a command like this:

```
RESTORE DATABASE MonthOfLunches
WITH RECOVERY;
```

That will immediately complete the recovery process and the database will be online again. However, if I do that, I won't be able to apply the log backups. I need the database to be in that restoring state in order to apply one or more log backups. To apply the log, I have to use the following T-SQL statement:

```
RESTORE LOG MonthOfLunches
FROM DISK = N'C:\data\MonthOfLunches_LogBackup_2011-12-27_10-29-42.bak'
WITH RECOVERY;
```

This will apply the transactions that are stored in the log backup and it will recover the database. The command should be clear. We're going to RESTORE but, this time, a LOG. It's going to be applied to the database MonthOfLunches. Everything else is the same as if we were doing a database restore. But, we want to stop at a particular point in time. To do that, we would use this command:

```
RESTORE LOG MonthOfLunches
FROM DISK = N'C:\data\MonthOfLunches_LogBackup_2011-12-27_10-29-42.bak'
WITH STOPAT = N'2011-12-27T09:15:00', NORECOVERY ;
```

For Source Code, Sample Chapters, the Author Forum and other resources, go to  
<http://www.manning.com/fritchey/>

Now what will happen is that the log will be applied to the backup. If, within this log, the transactions stop at the value specified, 2011-12-27T09:15:00, then we'll need to issue the `RECOVERY` command for the database. If not, you'll have to keep applying log backups until you arrive at the appropriate time.

It's very important that you have all the log backups between the full backup and the point in time you're going to recover to. If you do not have all of those log backups, you will get an error. The log backups must also be applied in order. You can't skip around. While this process is not all hard, it can be very tedious.

### **Summary**

While there are some rules around how you can run a restore, it's actually a very simple operation. But, as simple as it might be, it's worth practicing on a regular basis in order to be prepared in the event of an emergency. I showed you how you can restore log backups and get to a particular point in time using those log backups.

**Here are some other Manning titles you might be interested in:**



[SQL Server MVP Deep Dives, Volume 2](#)

Kalen Delaney, Louis Davidson, Greg Low, Brad McGehee, Paul Nielsen, Paul Randal, and Kimberly Tripp



[SQL Server DMVs in Action](#)

Ian W. Stirk



[ASP.NET MVC in Action](#)

Jeffrey Palermo, Ben Scheirman, and Jimmy Bogard

Last updated: May 1, 2012