

Placing Indexes on Filegroups

Source	http://msdn2.microsoft.com/en-us/library/aa178589(SQL.80).aspx
---------------	---

Placing Indexes on Filegroups

By default, indexes are created on the same filegroup as the base table on which the index is created. However, it is possible to create nonclustered indexes on a filegroup other than the filegroup of the base table. By creating the index on a different filegroup, you can realize performance gains if the filegroups make use of different physical drives with their own controllers. Data and index information can then be read in parallel by multiple disk heads. For example if **Table_A** on filegroup **f1** and **Index_A** on filegroup **f2** are both being used by the same query, performance gains can be achieved because both filegroups are being fully used with no contention. However, if **Table_A** is scanned by the query but **Index_A** is not referenced, only filegroup **f1** is used, resulting in no performance gain.

Because you cannot predict what type of access will take place and when it will take place, it could be a safer decision to spread your tables and indexes across all filegroups. This would guarantee that all disks are being accessed since all data and indexes are spread evenly across all disks, no matter which way the data is accessed. This is also a simpler approach for system administrators.

If there is a clustered index on a table, the data and the clustered index always reside in the same filegroup. Therefore, you can move a table from one filegroup to another by creating a clustered index on the base table that specifies a different filegroup on which to create the index (the index can then be dropped, leaving the base table in the new filegroup).

If the indexes of a table span multiple filegroups, all filegroups containing the table and its indexes must be backed up together, after which a transaction log backup must be created. Otherwise, only some of the indexes may be backed up, preventing the index from being recovered if the backup is restored later. For more information, see [Using File Backups](#).

Note An individual table or index can belong to only one filegroup; it cannot span filegroups.

~~~ End of Article ~~~