

---

## failover

The ability to automatically switch to a standby server in the event of a failure. In the MySQL context, failover involves a standby database server. Often supported within **J2EE** environments by the application server or framework.

See Also [Connector/J](#), [J2EE](#).

## Fast Index Creation

A capability first introduced in the InnoDB Plugin, now part of MySQL in 5.5 and higher, that speeds up creation of **InnoDB secondary indexes** by avoiding the need to completely rewrite the associated table. The speedup applies to dropping secondary indexes also.

Because index maintenance can add performance overhead to many data transfer operations, consider doing operations such as `ALTER TABLE ... ENGINE=INNODB` or `INSERT INTO ... SELECT * FROM ...` without any secondary indexes in place, and creating the indexes afterward.

In MySQL 5.6, this feature becomes more general. You can read and write to tables while an index is being created, and many more kinds of `ALTER TABLE` operations can be performed without copying the table, without blocking **DML** operations, or both. Thus in MySQL 5.6 and higher, this set of features is referred to as **online DDL** rather than Fast Index Creation.

For related information, see [Section 15.12, “InnoDB and Online DDL”](#).

See Also [DML](#), [index](#), [online DDL](#), [secondary index](#).

## fast shutdown

The default **shutdown** procedure for **InnoDB**, based on the configuration setting `innodb_fast_shutdown=1`. To save time, certain **flush** operations are skipped. This type of shutdown is safe during normal usage, because the flush operations are performed during the next startup, using the same mechanism as in **crash recovery**. In cases where the database is being shut down for an upgrade or downgrade, do a **slow shutdown** instead to ensure that all relevant changes are applied to the **data files** during the shutdown.

See Also [crash recovery](#), [data files](#), [flush](#), [shutdown](#), [slow shutdown](#).

## file format

The file format for **InnoDB** tables.

See Also [file-per-table](#), [.ibd file](#), [ibdata file](#), [row format](#).

## file-per-table

A general name for the setting controlled by the `innodb_file_per_table` option, which is an important configuration option that affects aspects of **InnoDB** file storage, availability of features, and I/O characteristics. As of MySQL 5.6.7, `innodb_file_per_table` is enabled by default.

With the `innodb_file_per_table` option enabled, you can create a table in its own **.ibd file** rather than in the shared **ibdata files** of the **system tablespace**. When table data is stored in an individual **.ibd file**, you have more flexibility to choose **row formats** required for features such as data **compression**. The `TRUNCATE TABLE` operation is also faster, and reclaimed space can be used by the operating system rather than remaining reserved for **InnoDB**.

The **MySQL Enterprise Backup** product is more flexible for tables that are in their own files. For example, tables can be excluded from a backup, but only if they are in separate files. Thus, this setting is suitable for tables that are backed up less frequently or on a different schedule.

See Also [compressed row format](#), [compression](#), [file format](#), [.ibd file](#), [ibdata file](#), [innodb\\_file\\_per\\_table](#), [MySQL Enterprise Backup](#), [row format](#), [system tablespace](#).

## fill factor

In an **InnoDB index**, the proportion of a **page** that is taken up by index data before the page is split. The unused space when index data is first divided between pages allows for rows to be updated with longer string