Checksums

Starting in early 2012, metadata checksums were added to all major ext4 and jbd2 data structures. The associated feature flag is metadata_csum. The desired checksum algorithm is indicated in the superblock, though as of October 2012 the only supported algorithm is crc32c. Some data structures did not have space to fit a full 32-bit checksum, so only the lower 16 bits are stored. Enabling the 64bit feature increases the data structure size so that full 32-bit checksums can be stored for many data structures. However, existing 32-bit filesystems cannot be extended to enable 64bit mode, at least not without the experimental resize2fs patches to do so.

Existing filesystems can have checksumming added by running `tune2fs -O metadata_csum` against the underlying device. If tune2fs encounters directory blocks that lack sufficient empty space to add a checksum, it will request that you run `e2fsck -D` to have the directories rebuilt with checksums. This has the added benefit of removing slack space from the directory files and rebalancing the htree indexes. If you _ignore_ this step, your directories will not be protected by a checksum!

The following table describes the data elements that go into each type of checksum. The checksum function is whatever the superblock describes (crc32c as of October 2013) unless noted otherwise.

Metadata	**Length**	**Ingredients**
Superblock	__le32	The entire superblock up to the checksum field. The UUID lives inside the superblock.
MMP	__le32	UUID + the entire MMP block up to the checksum field.
Extended Attributes	__le32	UUID + the entire extended attribute block. The checksum field is set to zero.
Directory Entries	__le32	UUID + inode number + inode generation + the directory block up to the fake entry enclosing the checksum field.
HTREE Nodes	__le32	UUID + inode number + inode generation + all valid extents + HTREE tail. The checksum field is set to zero.
Extents	__le32	UUID + inode number + inode generation + the entire extent block up to the checksum field.
Bitmaps	__le32 or __le16	UUID + the entire bitmap. Checksums are stored in the group descriptor, and truncated if the group descriptor size is 32 bytes (i.e. ^64bit)
Inodes	__le32	UUID + inode number + inode generation + the entire inode. The checksum field is set to zero. Each inode has its own checksum.
Group Descriptors	__le16	If metadata_csum, then UUID + group number + the entire descriptor; else if gdt_csum, then crc16(UUID + group number + the entire descriptor). In all cases, only the lower 16 bits are stored.