Merged Response

`btrfs check --repair` : Comprehensive Guide

Introduction

The `btrfs check --repair` command is a powerful filesystem recovery tool designed to check and repair corruption in Btrfs (B-tree File System) filesystems o

Important Safety Precautions

CRITICAL WARNING: This command carries significant risks:

- **Unmount First**: Always run this command on an **unmounted** filesystem to prevent further corruption
- **Create Backups**: Back up all critical data before running this command
- **Last Resort Option**: Consider this a last-resort recovery method after trying safer alternatives
- **Data Loss Risk**: Repairs may lead to partial or complete data loss, especially in cases of severe corruption
- **No Guarantees**: The repair process is not guaranteed to succeed and may sometimes worsen corruption

When to Use

This command is appropriate when:

- Your Btrfs filesystem refuses to mount due to metadata inconsistencies
- System logs show corruption errors like "corrupt leaf," "checksum errors," or "failed to read chunk tree"
- You've already attempted safer recovery methods like mounting with the recovery option
- You have no other means to recover important data and are aware of the risks

Basic Usage

```
1 btrfs check --repair /dev/sdX
```

Replace `/dev/sdX` with the appropriate device path (e.g., `/dev/sda2`).

Additional Parameters

Safety and Control Options

- `--readonly` or `-r` : Check without modifying anything (diagnostics only)
- `--force` : Override safety checks (use with extreme caution)
- `--progress` : Show progress during long operations (recommended for large filesystems)
- `-v` or `--verbose` : Increase output verbosity
- `-q` or `--quiet` : Suppress output except for errors

Alternative Root Options

- `--super=N` : Use alternative superblock copy (0, 1, or 2)
- `--backup` : Use backup root tree
- `--tree-root=BYTENR` : Specify alternative tree root
- `--chunk-root=BYTENR` : Specify alternative chunk root

Repair Strategies

- `--init-csum-tree` : Rebuild checksum tree from scratch
- `--init-extent-tree` : Rebuild extent tree from scratch (useful for severe corruption)
- `--chunk-recover` : Recover chunk tree by scanning devices
- `--clear-space-cache` : Clear space cache during repair

Performance Options

- `--mode=MODE` : Set checking mode (lowmem, original, chunk-tree)
- `--check-lowmem` : Force low memory mode for large filesystems

Thoroughness Options

- `--check-data-csum` : Verify data checksums (much slower but more thorough)
- `--extra-extent-checking` : Perform deep check of extent references
- `--qgroup-report` : Report quota group inconsistencies
- `--subvol-extents` : Check subvolume metadata

Recommended Recovery Workflow

1. **Attempt Safer Methods First**:

```
1 mount -o recovery,ro /dev/sdX /mnt
```

Try to recover data using the recovery mount option first

2. **Diagnostic Check**:

```
btrfs check /dev/sdX
```

Perform an initial check without repair to assess damage

3. **Create Backups** (if possible):

```
# If you can mount read-only
mount -o ro,recovery /dev/sdX /mnt
cp -a /mnt /path/to/backup/location
umount /mnt

```

4. **Unmount Filesystem**:

```
umount /path/to/mount
```

Ensure the filesystem is completely unmounted

5. **Attempt Repair**:

```
btrfs check --repair --progress /dev/sdX
```

Use progress flag for visibility during long operations

6. **Targeted Repair** (if initial repair fails):

```
btrfs check --repair --init-extent-tree /dev/sdX
```

Based on error messages, you may need specific repair options

7. **Verification**:

```
# After successful repair
mount /dev/sdX /mnt
btrfs scrub start /mnt

```

Run a scrub to verify integrity after repair

Alternative Recovery Tools

If `btrfs check --repair` fails or seems too risky, consider these alternatives:

- `btrfs rescue super-recover /dev/sdX` : Attempts to recover the superblock
- `btrfs rescue chunk-recover /dev/sdX` : Attempts to recover the chunk tree
- `btrfs restore /dev/sdX /path/to/recovery` : Extract files from a damaged filesystem
- `btrfs scrub start /mount/point` : Check and repair data errors (works on mounted filesystems)

Important Considerations

- **Tool Version**: Ensure you're using a recent version of btrfs-progs; older versions may have bugs or less effective repair algorithms
- **System Recovery**: For system volumes, boot from a live USB/recovery media
- **Hardware Issues**: First rule out hardware problems with tools like `smartctl` before attempting filesystem repairs
- **Multi-device Volumes**: For Btrfs RAID configurations, additional complexities may arise

Conclusion

The `btrfs check --repair` command is a powerful but potentially destructive recovery tool. While it can save your data in emergency situations, the best def