# OCFS2 file system - online file check

This document will describe OCFS2 online file check feature.

## Introduction

OCFS2 is often used in high-availability systems. However, OCFS2 usually converts the filesystem to read-only when encounters an error. This may not be necessary, since turning the filesystem read-only would affect other running processes as well, decreasing availability. Then, a mount option (errors=continue) is introduced, which would return the -EIO errno to the calling process and terminate further processing so that the filesystem is not corrupted further. The filesystem is not converted to read-only, and the problematic file's inode number is reported in the kernel log. The user can try to check/fix this file via online filecheck feature.

## Scope

This effort is to check/fix small issues which may hinder day-to-day operations of a cluster filesystem by turning the filesystem read-only. The scope of checking/fixing is at the file level, initially for regular files and eventually to all files (including system files) of the filesystem.

In case of directory to file links is incorrect, the directory inode is reported as erroneous.

This feature is not suited for extravagant checks which involve dependency of other components of the filesystem, such as but not limited to, checking if the bits for file blocks in the allocation has been set. In case of such an error, the offline fsck should/would be recommended.

Finally, such an operation/feature should not be automated lest the filesystem may end up with more damage than before the repair attempt. So, this has to be performed using user interaction and consent.

## User interface

When there are errors in the OCFS2 filesystem, they are usually accompanied by the inode number which caused the error. This inode number would be the input to check/fix the file.

There is a sysfs directory for each OCFS2 file system mounting:

```
/sys/fs/ocfs2/<devname>/filecheck
```

Here, <devname> indicates the name of OCFS2 volume device which has been already mounted. The file above would accept inode numbers. This could be used to communicate with kernel space, tell which file(inode number) will be checked or fixed. Currently, three operations are supported, which includes checking inode, fixing inode and setting the size of result record history.

1. If you want to know what error exactly happened to <inode> before fixing, do:

```
# echo "<inode>" > /sys/fs/ocfs2/<devname>/filecheck/check
# cat /sys/fs/ocfs2/<devname>/filecheck/check
```

The output is like this:

```
INO         DONE    ERROR
39502               1       GENERATION

<INO> lists the inode numbers.
<DONE> indicates whether the operation has been finished.
<ERROR> says what kind of errors was found. For the detailed error numbers,
please refer to the file linux/fs/ocfs2/filecheck.h.
```

2. If you determine to fix this inode, do:

```
# echo "<inode>" > /sys/fs/ocfs2/<devname>/filecheck/fix
# cat /sys/fs/ocfs2/<devname>/filecheck/fix
```

The output is like this::

```
INO         DONE    ERROR
39502               1       SUCCESS
```

This time, the <ERROR> column indicates whether this fix is successful or not.

3. The record cache is used to store the history of check/fix results. It's default size is 10, and can be adjust between the range of 10 ~ 100. You can adjust the size like this:

```
# echo "<size>" > /sys/fs/ocfs2/<devname>/filecheck/set
```

## Fixing stuff

On receiving the inode, the filesystem would read the inode and the file metadata. In case of errors, the filesystem would fix the errors and report the problems it fixed in the kernel log. As a precautionary measure, the inode must first be checked for errors before performing a final fix.

The inode and the result history will be maintained temporarily in a small linked list buffer which would contain the last (N) inodes fixed/checked, the detailed errors which were fixed/checked are printed in the kernel log.