Title

Read only dos attribute behavior on chmod 444/644

Reference Numbers

FITS:	-
PMR:	53289,003,756
CQ:	-
Complain:	AH6952

Description

Mobileye is using Bitkeeper as SCM. Some developers use Visual Studio to edit projects that are compiled and tested on Unix. The work flow currently shares the same files with SMB and NFS. The former is used to edit the project files with Visual Studio, the later is used to run Bitkeeper commands and to compile and test the written code.
The work flow relies on the Visual Studio users getting a warning that they are editing files that are not checked out on Bitkeeper.
The users are accustomed to a lock symbol being displayed in the tab holding the file. They are also used to a particular warning message when they attempt to write the file. There is no warning message when they attempt to edit a file that's not checked out. So the users are probably accustomed to pay attention to the lock symbol to avoid issues with merging changes into files that are checked out late.
Both, the lock and the particular warning message are triggered by the read only dos attribute. So the Mobileye work flow depends on a read only dos attribute being set on files that are not checked out and the attribute being cleared on checked out files. Since Bitkeeper does a chmod 444 on files that are not checked out and a chmod 644 on checked out files that NFS call will have to toggle the read only dos attribute in their work flow.
There is apparently no trivial way to configure Visual Studio to rely on the ACLs instead of the read only dos attribute. If Visual Studio can't modify a file it will attempt to delete the file and offer to save it again. The file is already deleted by Visual Studio at the time it offers to save it. This is possible due to the delete child permission being typically set on the parent folders.
To keep the accustomed tools and work flow Mobileye requires the read only dos attribute being set on files after a chmod 444 and unset after a chmod 644.

Work around

Work around description

For the customer use case it's apparently sufficient to always determine the state of the read only dos attribute based on the w permission in the mode bits of the owner.
It might be useful if toggling the read only dos attribute also toggles the w permission of the mode bits of the owner since the previous NAS displayed a similar behavior.
The other dos attributes are not important for this particular work flow but might be important for other applications in the company.
Also the NTFS ACLs are not relevant for this particular work flow which only relies on the read only dos attribute and the Posix permissions due to the sharing of files through NFS and the frequent chmod by Bitkeeper.

The proposed workaround will tie the read only dos attribute to the owner write permission of the Posix mode bits. Since the mode bits are tied to the ACLs a change in the read only dos attribute will also toggle the write permission in the ACL entry of the file owner. This toggle operation triggers a chmod that will drop all ACL entries that are not part of the Posix mode bits.
The other dos attributes will not be mapped to mode bits and they will also not be stored.

Due to these limitations it's recommended to apply this workaround on a per share base. If necessary it's recommended to create a special share for the development with Visual Studio. Such a share should also be created for testing this workaround with the particular Visual Studio versions in use at the customer.

To enable the workaround on a share the following commands have to be executed on one node of the V7KU system as root:

```
net conf setparm <sharename> 'store dos attributes' no
net conf setparm <sharename> 'map readonly' yes
net conf setparm <sharename> 'map archive' no
net conf setparm <sharename> 'map hidden' no
net conf setparm <sharename> 'map system' no
```

Tod disable the workaround the following commands have to be executed on one node of the V7KU system as root:

```
net conf delparm <sharename> 'store dos attributes'
net conf delparm <sharename> 'map readonly'
net conf delparm <sharename> 'map archive'
net conf delparm <sharename> 'map hidden'
net conf delparm <sharename> 'map system'
```

These changes will apply to new connections made by clients to these shares.

Dependencies: none

Implications and side effect:

With this workaround the dos attributes hidden, system, archive, and sparse can't be stored anymore and will always be unset on the shares with the workaround enabled.
This might confuse backup applications relying on the archive bit. Most of todays backup applications including TSM are not affected.
Without the system bit being set on home folders windows doesn't read the desktop.ini in the folders to apply a custom icon to the home folder.
Changing the read only dos attribute will drop ACL entries that are not related to the Posix mode bits.

Service & upgrade implications: none

Test:

The following actions have been performed to verify this function.

The automated BVT regression test suite has been run successfully on a system with these settings. All found issues are directly related to the changed behavior of the dos attributes with no unexpected side effects.

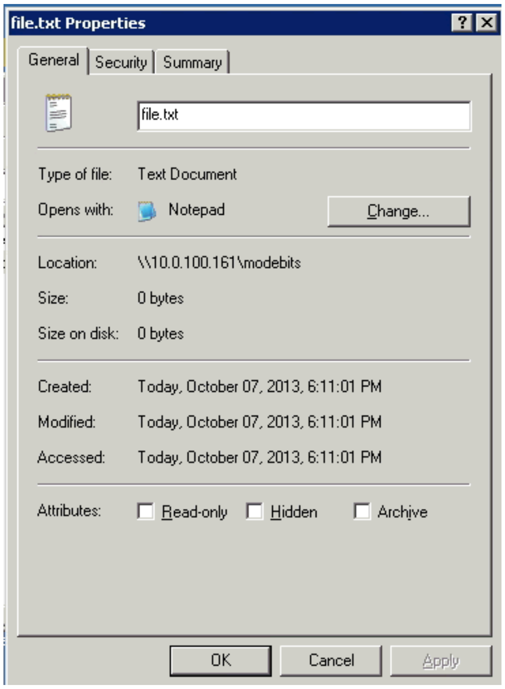
To test the read only dos attribute behavior the following steps have been executed:
Create a file and set the file owner:

```
touch /ibm/gpfs0/abc/modebits/file.txt
chown 'virtual1\testuser1' /ibm/gpfs0/abc/modebits/file.txt
```

Check the current state of the mode bits and dos attributes:

```
 # ls -l /ibm/gpfs0/abc/modebits/file.txt
```

```
-rw-r--r-- 1 VIRTUAL1\testuser1 root 0 Oct  7 19:11 /ibm/gpfs0/abc/modebits/file.txt
```



file.txt

Type of file: Text Document

Opens with: Notepad

Location: \\10.0.100.161\modebits

Size: 0 bytes

Size on disk: 0 bytes

Created: Today, October 07, 2013, 6:11:01 PM

Modified: Today, October 07, 2013, 6:11:01 PM

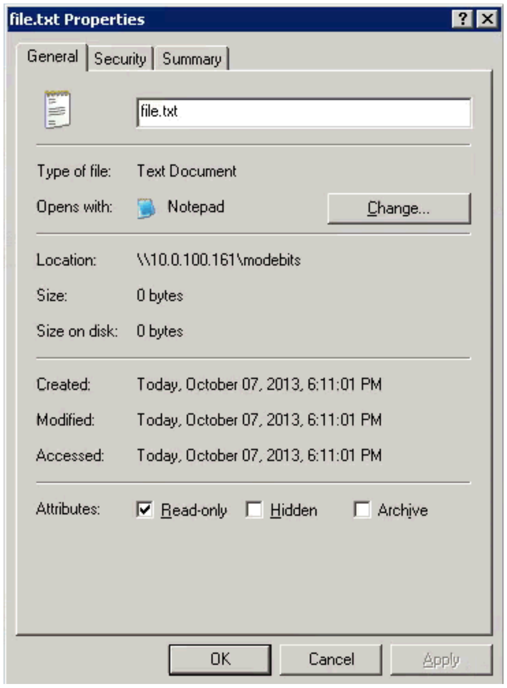
Accessed: Today, October 07, 2013, 6:11:01 PM

Attributes: ☐ Read-only ☐ Hidden ☐ Archive

Trigger a chmod system call and check the mode bits and dos attributes again:

```
# chmod 444 /ibm/gpfs0/abc/modebits/file.txt

# ls -l /ibm/gpfs0/abc/modebits/file.txt
-r--r--r-- 1 VIRTUAL1\testuser1 root 0 Oct  7 19:11 /ibm/gpfs0/abc/modebits/file.txt
```



file.txt

Type of file: Text Document

Opens with: Notepad

Location: \\10.0.100.161\modebits

Size: 0 bytes

Size on disk: 0 bytes

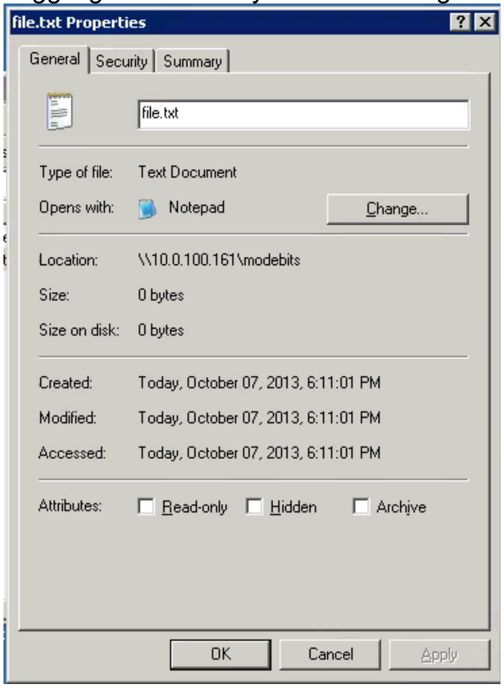
Created: Today, October 07, 2013, 6:11:01 PM

Modified: Today, October 07, 2013, 6:11:01 PM

Accessed: Today, October 07, 2013, 6:11:01 PM

Attributes: ☑ Read-only ☐ Hidden ☐ Archive

Toggling the read only attribute through the Windows GUI will change the mode bits:



file.txt

Type of file: Text Document

Opens with: Notepad

Location: \\10.0.100.161\modebits

Size: 0 bytes

Size on disk: 0 bytes

Created: Today, October 07, 2013, 6:11:01 PM

Modified: Today, October 07, 2013, 6:11:01 PM

Accessed: Today, October 07, 2013, 6:11:01 PM

Attributes: □ Read-only □ Hidden □ Archive

```
# ls -l /ibm/gpfs0/abc/modebits/file.txt
-rw-r--r-- 1 VIRTUAL1\testuser1 root 0 Oct  7 19:11 /ibm/gpfs0/abc/modebits/file.txt
```

To successfully change the read only dos attribute through the windows GUI it's required to be the file owner.