

Bug in XP SP2: RtlFindLastBackwardRunClear

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2008-01/msg00248.html>

- *From:* "Maxim S. Shatskih" <maxim@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 14 Jan 2008 09:38:19 +0300
-

This code:

```
ULONG Start;  
RtlSetBits(Bitmap, 0, 100);  
(VOID)RtlFindLastBackwardRunClear(Bitmap, 3, &Start);
```

causes AV fault inside RtlFindLastBackwardRunClear. This is surely a bug, since 3 is a valid value for StartingBitPos, and &Start pointer is also valid. The pointed value is not relevant, since it is OUT parameter. So, RtlFindLastBackwardRunClear is called correctly.

The fault is at:

```
mov edx,dword ptr [esi+edx*4]
```

where esi is RTL_BITMAP::Buffer, and edx is shifted right by 5 (32 bits in ULONG?) just above this instruction. edx value is 07fffffe, so, before the shift, it was FFFFFFFC0.

Obviously some variable wrapped below zero.

What is really funny: I'm porting my old driver from NT4 DDK to WDK. It is time to upgrade the build env.

NT4 had no RtlFindLastBackwardRunClear, so, several years ago I wrote my own implementation of this routine, and it worked fine. On a unit test suite in a WDK build, it gives the same results as RtlFindLastBackwardRunClear, but does not crash on small (I expect this occurs if StartingBitPos is < 32 or <= 32, but I'm not sure of it) values of StartingBitPos.

Surely I will switch back to my code, but it's a pity that the Rtl routine in the kernel has such a bug. Looks like yet another MmGetSystemRoutineAddress issue.

—
Maxim Shatskih, Windows DDK MVP
StorageCraft Corporation
maxim@xxxxxxxxxxxxxxxxxxxx

Bug in XP SP2: RtlFindLastBackwardRunClear

<http://www.storagecraft.com>