

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-05/msg09232.html>

- *From:* Anton Altaparmakov <aia21@xxxxxxxxxx>
 - *Date:* Mon, 21 May 2007 09:37:42 +0100 (BST)
-

Hi Linus/Andrew,

Please apply the below patch. It fixes the LDM driver so that it works with Windows Vista dynamic disks which are subtly different to Windows 2000/XP ones.

This patch was needed to get a Vista formatted dynamic disk to be recognized and parsed successfully.

Thanks go to Chris Teachworth for the report and testing.

Best regards,

Anton

--

Anton Altaparmakov <aia21 at cam.ac.uk> (replace at with @)
Unix Support, Computing Service, University of Cambridge, CB2 3QH, UK
Linux NTFS maintainer, <http://www.linux-ntfs.org/>

From: Anton Altaparmakov <aia21@xxxxxxxxxx>

Fix the LDM driver to work with Windows Vista dynamic disks.

Signed-off-by: Anton Altaparmakov <aia21@xxxxxxxxxx>

Cc: Richard Russon <ldm@xxxxxxxxxx>

diff --git a/Documentation/ldm.txt b/Documentation/ldm.txt

index e266e11..718085b 100644

--- a/Documentation/ldm.txt

+++ b/Documentation/ldm.txt

@@ -2,10 +2,13 @@

LDM – Logical Disk Manager (Dynamic Disks)

+Originally Written by FlatCap – Richard Russon <ldm@xxxxxxxxxx>.

+Last Updated by Anton Altaparmakov on 30 March 2007 for Windows Vista.

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

+
Overview

-Windows 2000 and XP use a new partitioning scheme. It is a complete
+Windows 2000, XP, and Vista use a new partitioning scheme. It is a complete replacement for the MSDOS style partitions. It stores its information in a 1MiB journalled database at the end of the physical disk. The size of partitions is limited only by disk space. The maximum number of partitions is @@ -23,7 +26,11 @@ Once the LDM driver has divided up the disk, you can use the MD driver to assemble any multi-partition volumes, e.g. Stripes, RAID5.

To prevent legacy applications from repartitioning the disk, the LDM creates a -dummy MSDOS partition containing one disk-sized partition.

+dummy MSDOS partition containing one disk-sized partition. This is what is +supported with the Linux LDM driver.

+
+A newer approach that has been implemented with Vista is to put LDM on top of a +GPT label disk. This is not supported by the Linux LDM driver yet.

Example

@@ -88,13 +95,13 @@ and cannot boot from a Dynamic Disk.

More Documentation

-There is an Overview of the LDM online together with complete Technical
-Documentation. It can also be downloaded in html.
+There is an Overview of the LDM together with complete Technical Documentation.
+It is available for download.

- <http://linux-ntfs.sourceforge.net/ldm/index.html>

- <http://linux-ntfs.sourceforge.net/downloads.html>

+ <http://www.linux-ntfs.org/content/view/19/37/>

-If you have any LDM questions that aren't answered on the website, email me.

+If you have any LDM questions that aren't answered in the documentation, email

+me.

Cheers,

FlatCap - Richard Russon

diff --git a/MAINTAINERS b/MAINTAINERS

index 4c3277c..22ab401 100644

--- a/MAINTAINERS

+++ b/MAINTAINERS

@@ -2231,11 +2231,11 @@ M: khali@xxxxxxxxxxxxx

L: lm-sensors@xxxxxxxxxxxxxx

S: Maintained

-LOGICAL DISK MANAGER SUPPORT (LDM, Windows 2000/XP Dynamic Disks)

+LOGICAL DISK MANAGER SUPPORT (LDM, Windows 2000/XP/Vista Dynamic Disks)

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

P: Richard Russon (FlatCap)
M: ldm@xxxxxxxxxxxxx
-L: ldm-devel@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
-W: <http://ldm.sourceforge.net>
+L: linux-ntfs-dev@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx
+W: <http://www.linux-ntfs.org/content/view/19/37/>
S: Maintained

```
LSILOGIC MPT FUSION DRIVERS (FC/SAS/SPI)
diff --git a/fs/partitions/Kconfig b/fs/partitions/Kconfig
index 7638a1c..a99acd8 100644
--- a/fs/partitions/Kconfig
+++ b/fs/partitions/Kconfig
@@ -166,8 +166,12 @@ config LDM_PARTITION
depends on PARTITION_ADVANCED
---help---
```

Say Y here if you would like to use hard disks under Linux which

- were partitioned using Windows 2000's or XP's Logical Disk Manager.
- They are also known as "Dynamic Disks".
- + were partitioned using Windows 2000's/XP's or Vista's Logical Disk Manager. They are also known as "Dynamic Disks".

+
+ Note this driver only supports Dynamic Disks with a protective MBR label, i.e. DOS partition table. It does not support GPT labelled Dynamic Disks yet as can be created with Vista.

Windows 2000 introduced the concept of Dynamic Disks to get around the limitations of the PC's partitioning scheme. The Logical Disk

```
@@ -175,8 +179,8 @@ config LDM_PARTITION
```

mirrored, striped or RAID volumes, all without the need for rebooting.

- Normal partitions are now called Basic Disks under Windows 2000 and XP.
- + Normal partitions are now called Basic Disks under Windows 2000, XP, and Vista.

For a fuller description read <file:Documentation/ldm.txt>.

```
diff --git a/fs/partitions/ldm.c b/fs/partitions/ldm.c
index 1a60926..c387812 100644
--- a/fs/partitions/ldm.c
+++ b/fs/partitions/ldm.c
@@ -2,10 +2,10 @@
* ldm - Support for Windows Logical Disk Manager (Dynamic Disks)
*
* Copyright (C) 2001,2002 Richard Russon <ldm@xxxxxxxxxxxxx>
- * Copyright (c) 2001-2004 Anton Altaparmakov
+ * Copyright (c) 2001-2007 Anton Altaparmakov
* Copyright (C) 2001,2002 Jakob Kemi <jakob.kemi@xxxxxxxxxx>
*
```

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

```
- * Documentation is available at http://linux-ntfs.sf.net/ldm
+ * Documentation is available at http://www.linux-ntfs.org/content/view/19/37/
*
* This program is free software; you can redistribute it and/or modify it under
* the terms of the GNU General Public License as published by the Free Software
@@ -62,7 +62,6 @@ static void _ldm_printk (const char *level, const char *function,
printk ("%s%s(): %s\n", level, function, buf);
}

-
-/**
* ldm_parse_hexbyte - Convert a ASCII hex number to a byte
* @src: Pointer to at least 2 characters to convert.
@@ -118,7 +117,6 @@ static bool ldm_parse_guid (const u8 *src, u8 *dest)
return true;
}

-
-/**
* ldm_parse_privhead - Read the LDM Database PRIVHEAD structure
* @data: Raw database PRIVHEAD structure loaded from the device
@@ -130,46 +128,48 @@ static bool ldm_parse_guid (const u8 *src, u8 *dest)
* Return: 'true' @ph contains the PRIVHEAD data
* 'false' @ph contents are undefined
*/
-static bool ldm_parse_privhead (const u8 *data, struct privhead *ph)
+static bool ldm_parse_privhead(const u8 *data, struct privhead *ph)
{
- BUG_ON (!data || !ph);
+ bool is_vista = false;

- if (MAGIC_PRIVHEAD != BE64 (data)) {
- ldm_error ("Cannot find PRIVHEAD structure. LDM database is"
+ BUG_ON(!data || !ph);
+ if (MAGIC_PRIVHEAD != BE64(data)) {
+ ldm_error("Cannot find PRIVHEAD structure. LDM database is"
" corrupt. Aborting.");
return false;
}

-
- ph->ver_major = BE16 (data + 0x000C);
- ph->ver_minor = BE16 (data + 0x000E);
- ph->logical_disk_start = BE64 (data + 0x011B);
- ph->logical_disk_size = BE64 (data + 0x0123);
- ph->config_start = BE64 (data + 0x012B);
- ph->config_size = BE64 (data + 0x0133);
-
- if ((ph->ver_major != 2) || (ph->ver_minor != 11)) {
- ldm_error ("Expected PRIVHEAD version %d.%d, got %d.%d."
- " Aborting.", 2, 11, ph->ver_major, ph->ver_minor);
+ ph->ver_major = BE16(data + 0x000C);
```

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

```
+ ph->ver_minor = BE16(data + 0x000E);
+ ph->logical_disk_start = BE64(data + 0x011B);
+ ph->logical_disk_size = BE64(data + 0x0123);
+ ph->config_start = BE64(data + 0x012B);
+ ph->config_size = BE64(data + 0x0133);
+ /* Version 2.11 is Win2k/XP and version 2.12 is Vista. */
+ if (ph->ver_major == 2 && ph->ver_minor == 12)
+ is_vista = true;
+ if (!is_vista && (ph->ver_major != 2 || ph->ver_minor != 11)) {
+ ldm_error("Expected PRIVHEAD version 2.11 or 2.12, got %d.%d."
+ " Aborting.", ph->ver_major, ph->ver_minor);
return false;
}
+ ldm_debug("PRIVHEAD version %d.%d (Windows %s).", ph->ver_major,
+ ph->ver_minor, is_vista ? "Vista" : "2000/XP");
if (ph->config_size != LDM_DB_SIZE) { /* 1 MiB in sectors. */
- /* Warn the user and continue, carefully */
- ldm_info ("Database is normally %u bytes, it claims to "
+ /* Warn the user and continue, carefully. */
+ ldm_info("Database is normally %u bytes, it claims to "
"be %llu bytes.", LDM_DB_SIZE,
- (unsigned long long)ph->config_size );
+ (unsigned long long)ph->config_size);
}
- if ((ph->logical_disk_size == 0) ||
- (ph->logical_disk_start + ph->logical_disk_size > ph->config_start)) {
- ldm_error ("PRIVHEAD disk size doesn't match real disk size");
+ if ((ph->logical_disk_size == 0) || (ph->logical_disk_start +
+ ph->logical_disk_size > ph->config_start)) {
+ ldm_error("PRIVHEAD disk size doesn't match real disk size");
return false;
}
-
- if (!ldm_parse_guid (data + 0x0030, ph->disk_id)) {
- ldm_error ("PRIVHEAD contains an invalid GUID.");
+ if (!ldm_parse_guid(data + 0x0030, ph->disk_id)) {
+ ldm_error("PRIVHEAD contains an invalid GUID.");
return false;
}
-
- ldm_debug ("Parsed PRIVHEAD successfully.");
+ ldm_debug("Parsed PRIVHEAD successfully.");
return true;
}

@@ -409,7 +409,7 @@ out:
* Return: 'true' @toc1 contains validated TOCBLOCK info
* 'false' @toc1 contents are undefined
*/
-static bool ldm_validate_tocblocks (struct block_device *bdev,
+static bool ldm_validate_tocblocks(struct block_device *bdev,
```

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

```
unsigned long base, struct ldmdb *ldb)
{
static const int off[4] = { OFF_TOCB1, OFF_TOCB2, OFF_TOCB3, OFF_TOCB4};
@@ -417,54 +417,57 @@ static bool ldm_validate_tocblocks (struct block_device *bdev,
struct privhead *ph;
Sector sect;
u8 *data;
+ int i, nr_tbs;
bool result = false;
- int i;

- BUG_ON (!bdev || !ldb);
-
- ph = &ldb->ph;
+ BUG_ON(!bdev || !ldb);
+ ph = &ldb->ph;
tb[0] = &ldb->toc;
- tb[1] = kmalloc (sizeof (*tb[1]), GFP_KERNEL);
- tb[2] = kmalloc (sizeof (*tb[2]), GFP_KERNEL);
- tb[3] = kmalloc (sizeof (*tb[3]), GFP_KERNEL);
- if (!tb[1] || !tb[2] || !tb[3]) {
- ldm_crit ("Out of memory.");
- goto out;
+ tb[1] = kmalloc(sizeof(*tb[1]) * 3, GFP_KERNEL);
+ if (!tb[1]) {
+ ldm_crit("Out of memory.");
+ goto err;
+ }
-
- for (i = 0; i < 4; i++) /* Read and parse all four toc's. */
- {
- data = read_dev_sector (bdev, base + off[i], &sect);
+ tb[2] = (struct tocblock*)((u8*)tb[1] + sizeof(*tb[1]));
+ tb[3] = (struct tocblock*)((u8*)tb[2] + sizeof(*tb[2]));
+ /*
+ * Try to read and parse all four TOCBLOCKS.
+ *
+ * Windows Vista LDM v2.12 does not always have all four TOCBLOCKS so
+ * skip any that fail as long as we get at least one valid TOCBLOCK.
+ */
+ for (nr_tbs = i = 0; i < 4; i++) {
+ data = read_dev_sector(bdev, base + off[i], &sect);
if (!data) {
- ldm_crit ("Disk read failed.");
- goto out;
+ ldm_error("Disk read failed for TOCBLOCK %d.", i);
+ continue;
+ }
- result = ldm_parse_tocblock (data, tb[i]);
- put_dev_sector (sect);
- if (!result)
```

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

```
- goto out; /* Already logged */
+ if (ldm_parse_tocblock(data, tb[nr_tbs]))
+ nr_tbs++;
+ put_dev_sector(sect);
}
-
- /* Range check the toc against a privhead. */
+ if (!nr_tbs) {
+ ldm_crit("Failed to find a valid TOCBLOCK.");
+ goto err;
+ }
+ /* Range check the TOCBLOCK against a privhead. */
if (((tb[0]->bitmap1_start + tb[0]->bitmap1_size) > ph->config_size) ||
- ((tb[0]->bitmap2_start + tb[0]->bitmap2_size) > ph->config_size)) {
- ldm_crit ("The bitmaps are out of range. Giving up.");
- goto out;
+ ((tb[0]->bitmap2_start + tb[0]->bitmap2_size) >
+ ph->config_size)) {
+ ldm_crit("The bitmaps are out of range. Giving up.");
+ goto err;
}
-
- if (!ldm_compare_tocblocks (tb[0], tb[1]) || /* Compare all tocs. */
- !ldm_compare_tocblocks (tb[0], tb[2]) ||
- !ldm_compare_tocblocks (tb[0], tb[3])) {
- ldm_crit ("The TOCBLOCKS don't match.");
- goto out;
+ /* Compare all loaded TOCBLOCKS. */
+ for (i = 1; i < nr_tbs; i++) {
+ if (!ldm_compare_tocblocks(tb[0], tb[i])) {
+ ldm_crit("TOCBLOCKS 0 and %d do not match.", i);
+ goto err;
+ }
}
-
- ldm_debug ("Validated TOCBLOCKS successfully.");
+ ldm_debug("Validated %d TOCBLOCKS successfully.", nr_tbs);
result = true;
-out:
- kfree (tb[1]);
- kfree (tb[2]);
- kfree (tb[3]);
+err:
+ kfree(tb[1]);
return result;
}

@@ -566,7 +569,7 @@ static bool ldm_validate_partition_table (struct block_device *bdev)

p = (struct partition*)(data + 0x01BE);
for (i = 0; i < 4; i++, p++)
```

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

```
- if (SYS_IND (p) == WIN2K_DYNAMIC_PARTITION) {
+ if (SYS_IND (p) == LDM_PARTITION) {
result = true;
break;
}
@@ -975,44 +978,68 @@ static bool ldm_parse_dsk4 (const u8 *buffer, int buflen, struct vblk *vb)
* Return: 'true' @vb contains a Partition VBLK
* 'false' @vb contents are not defined
*/
-static bool ldm_parse prt3 (const u8 *buffer, int buflen, struct vblk *vb)
+static bool ldm_parse prt3(const u8 *buffer, int buflen, struct vblk *vb)
{
int r_objid, r_name, r_size, r_parent, r_diskid, r_index, len;
struct vblk_part *part;

- BUG_ON (!buffer || !vb);
-
- r_objid = ldm_relative (buffer, buflen, 0x18, 0);
- r_name = ldm_relative (buffer, buflen, 0x18, r_objid);
- r_size = ldm_relative (buffer, buflen, 0x34, r_name);
- r_parent = ldm_relative (buffer, buflen, 0x34, r_size);
- r_diskid = ldm_relative (buffer, buflen, 0x34, r_parent);
-
+ BUG_ON(!buffer || !vb);
+ r_objid = ldm_relative(buffer, buflen, 0x18, 0);
+ if (r_objid < 0) {
+ ldm_error("r_objid %d < 0", r_objid);
+ return false;
+ }
+ r_name = ldm_relative(buffer, buflen, 0x18, r_objid);
+ if (r_name < 0) {
+ ldm_error("r_name %d < 0", r_name);
+ return false;
+ }
+ r_size = ldm_relative(buffer, buflen, 0x34, r_name);
+ if (r_size < 0) {
+ ldm_error("r_size %d < 0", r_size);
+ return false;
+ }
+ r_parent = ldm_relative(buffer, buflen, 0x34, r_size);
+ if (r_parent < 0) {
+ ldm_error("r_parent %d < 0", r_parent);
+ return false;
+ }
+ r_diskid = ldm_relative(buffer, buflen, 0x34, r_parent);
+ if (r_diskid < 0) {
+ ldm_error("r_diskid %d < 0", r_diskid);
+ return false;
+ }
if (buffer[0x12] & VBLK_FLAG_PART_INDEX) {
- r_index = ldm_relative (buffer, buflen, 0x34, r_diskid);
```

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

```
+ r_index = ldm_relative(buffer, buflen, 0x34, r_diskid);
+ if (r_index < 0) {
+ ldm_error("r_index %d < 0", r_index);
+ return false;
+ }
len = r_index;
} else {
r_index = 0;
len = r_diskid;
}
- if (len < 0)
+ if (len < 0) {
+ ldm_error("len %d < 0", len);
return false;
-
+ }
len += VBLK_SIZE_PRT3;
- if (len != BE32(buffer + 0x14))
+ if (len > BE32(buffer + 0x14)) {
+ ldm_error("len %d > BE32(buffer + 0x14) %d", len,
+ BE32(buffer + 0x14));
return false;
-
+ }
part = &vb->vblk.part;
- part->start = BE64(buffer + 0x24 + r_name);
- part->volume_offset = BE64(buffer + 0x2C + r_name);
- part->size = ldm_get_vnum(buffer + 0x34 + r_name);
- part->parent_id = ldm_get_vnum(buffer + 0x34 + r_size);
- part->disk_id = ldm_get_vnum(buffer + 0x34 + r_parent);
+ part->start = BE64(buffer + 0x24 + r_name);
+ part->volume_offset = BE64(buffer + 0x2C + r_name);
+ part->size = ldm_get_vnum(buffer + 0x34 + r_name);
+ part->parent_id = ldm_get_vnum(buffer + 0x34 + r_size);
+ part->disk_id = ldm_get_vnum(buffer + 0x34 + r_parent);
if (vb->flags & VBLK_FLAG_PART_INDEX)
part->partnum = buffer[0x35 + r_diskid];
else
part->partnum = 0;
-
return true;
}

@@ -1475,4 +1502,3 @@ out:
kfree(ldb);
return result;
}
-
diff --git a/fs/partitions/ldm.h b/fs/partitions/ldm.h
index 6e8d795..d2e6a30 100644
--- a/fs/partitions/ldm.h
```

[2.6 PATCH] LDM: Fix for Windows Vista dynamic disks

```
+++ b/fs/partitions/ldm.h
@@ -2,10 +2,10 @@
* ldm - Part of the Linux-NTFS project.
*
* Copyright (C) 2001,2002 Richard Russon <ldm@xxxxxxxxxxxx>
- * Copyright (C) 2001 Anton Altaparmakov <aia21@xxxxxxxxxxxx>
+ * Copyright (c) 2001-2007 Anton Altaparmakov
* Copyright (C) 2001,2002 Jakob Kemi <jakob.kemi@xxxxxxxxxx>
*
- * Documentation is available at http://linux-ntfs.sf.net/ldm
+ * Documentation is available at http://www.linux-ntfs.org/content/view/19/37/
*
* This program is free software; you can redistribute it and/or modify it
* under the terms of the GNU General Public License as published by the Free
@@ -93,7 +93,7 @@ struct parsed_partitions;

#define OFF_VMDB 17 /* List of partitions. */

-#define WIN2K_DYNAMIC_PARTITION 0x42 /* Formerly SFS (Landis). */
+#define LDM_PARTITION 0x42 /* Formerly SFS (Landis). */

#define TOC_BITMAP1 "config" /* Names of the two defined */
#define TOC_BITMAP2 "log" /* bitmaps in the TOCBLOCK. */
-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxxxx
More majordomo info at http://vger.kernel.org/majordomo-info.html
Please read the FAQ at http://www.tux.org/lkml/
```