

Table and List Test File

Dennis Grace, IBM

Table 1. Four column test table

| First column | Second column | Third column | Fourth column |
|---------------------|---|----------------------------|--|
| GUI | Filler | More filler and so forth | The frame attribute for this table is set to all, which previously produced a table complete with vertical separators. |
| Non-sequitur | Information lifted from another table: users who don't have GTK libraries or X Window Systems on their machines | All uses except automation | Allows you to choose from available options only, instead of having to sort through all the options, including ones that are not available at that point in the process. |
| Command Line | Expert | All uses | Allows easy automation of tasks |
| LVM Emulation Tools | Experienced LVM User | LVM manipulation | Allows users to keep old scripts and knowledge base with only slight changes. |

Variable list A. List format

Device Managers

The first layer is the device managers. These plug-ins communicate with the hardware device drivers to create the first EVMS objects. Currently, all devices are handled by a single plug-in. Future releases of EVMS might need additional device managers to do network device management, such as for disks on a storage area network (SAN).

Segment Managers

The second layer is the segment managers. These plug-ins handle the segmenting, or partitioning, of disk drives. The Engine components can replace partitioning programs, such as **fdisk** and **Disk Druid**, and the kernel components can replace the in-kernel disk partitioning code. Segment managers can also be "stacked," meaning that one segment manager can take as input output from another segment manager.

EVMS provides the following segment managers: DOS, GPT, and System/390® (S/390). Other segment manager plug-ins can be added to support other partitioning schemes.

Region Managers

The third layer is the region managers. This layer is intended to provide a place for plug-ins that ensure compatibility with existing volume management schemes in Linux or other operating systems. Region managers are intended to model systems that provide a logical abstraction above disks or partitions.

Like the segment managers, region managers can also be stacked. Therefore, the input object(s) to a region manager can be disks, segments, or other regions.

There are currently four region manager plug-ins in EVMS: Linux LVM, AIX, OS/2, and Multi-Disk (MD).

Nested variable list A.1

| | |
|------------------|---|
| Linux LVM | The Linux LVM plug-in provides compatibility with the Linux LVM and allows the creation of volume groups (known in EVMS as containers) and logical volumes (known in EVMS as regions). |
| AIX LVM | The AIX LVM provides compatibility with AIX and is similar in functionality to the Linux LVM by also using volume groups and logical volumes. |
| OS/2 LVM | The OS/2 plug-in provides compatibility with volumes created under OS/2. Unlike the Linux and AIX LVMS, the OS/2 LVM is based on linear linking of disk partitions, as well as bad-block relocation. The OS/2 LVM does not allow for modifications. |
| MD LVM | The Multi-Disk (MD) plug-in for RAID provides RAID levels linear, 0, 1, 4, and 5 in software. MD is one plug-in that will appear as four region managers you will be able to choose from. |

EVMS Features

The next layer is EVMS features. This layer is where new EVMS-native functionality is implemented. EVMS features can be built on any object in the system, including disks, segments, regions, or other feature objects. EVMS features all share a common type of metadata, which makes discovery of feature objects much more efficient, and recovery of broken features objects much more reliable. There are three features currently available in EVMS: Drive Linking, Bad Block Relocation, and Snapshotting.

Nested variable list A.2

| | |
|--------------------------------------|--|
| Drive Linking | The Drive Linking feature allows any number of objects to be linearly concatenated together into a single object. A Drive Linked volume can be expanded by adding another storage object to the end of the chain. |
| Bad Block Relocation | The Bad Block Relocation (BBR) feature monitors its I/O path and detects write failures (which can be caused by a damaged disk). In the event of such a failure, the data from that request is stored in a new location. BBR keeps track of this remapping. Additional I/Os to that location are redirected to the new location. |
| Snapshotting | The Snapshotting feature provides a mechanism for creating a "frozen" copy of a volume at a single instant in time, without having to take that volume off-line. This is useful for performing backups on a live system. Snapshots work with any volume (EVMS or compatibility), and can use any other available object as a backing store. After a snapshot is created and made into an EVMS volume, writes to the "original" volume cause the original contents of that location to be copied to the snapshot's storage object. Reads to the snapshot volume look like they come from the original at the time the snapshot was created. |
| File System Interface Modules | <p>File System Interface Modules (FSIMs) are the one layer of EVMS that exists only in user space. These plug-ins are used to provide coordination with the filesystems during certain volume management operations. For instance, when expanding or shrinking a volume, the filesystem must also be expanded or shrunk to the appropriate size. Ordering in this example is also important; a filesystem cannot be expanded before the volume, and a volume cannot be shrunk before the filesystem. The FSIMs allow EVMS to ensure this coordination and ordering.</p> <p>FSIMs also perform filesystem operations from one of the EVMS user interfaces. For instance, a user can make new filesystems and check existing filesystems by interacting with the FSIM.</p> |

The following list includes a modification to the list format to allow a block-type appearance. Note that the second list does not nest.

Variable list B. Block format.

Block list

The first layer is the device managers. These plug-ins communicate with the hardware device drivers to create the first EVMS objects. Currently, all devices are handled by a single plug-in. Future releases of EVMS might need additional device managers to do network device management, such as for disks on a storage area network (SAN).

Segment Managers

The second layer is the segment managers. These plug-ins handle the segmenting, or partitioning, of disk drives. The Engine components can replace partitioning programs, such as **fdisk** and **Disk Druid**, and the kernel components can replace the in-kernel disk partitioning code. Segment managers can also be "stacked," meaning that one segment manager can take as input output from another segment manager.

EVMS provides the following segment managers: DOS, GPT, and System/390® (S/390). Other segment manager plug-ins can be added to support other partitioning schemes.

Region Managers

The third layer is the region managers. This layer is intended to provide a place for plug-ins that ensure compatibility with existing volume management schemes in Linux or other operating systems. Region managers are intended to model systems that provide a logical abstraction above disks or partitions.

Like the segment managers, region managers can also be stacked. Therefore, the input object(s) to a region manager can be disks, segments, or other regions.

There are currently four region manager plug-ins in EVMS: Linux LVM, AIX, OS/2, and Multi-Disk (MD).

Nested variablelist B.1

First nested term

These terms should be nested. The Linux LVM plug-in provides compatibility with the Linux LVM and allows the creation of volume groups (known in EVMS as containers) and logical volumes (known in EVMS as regions).

Second nested term

The AIX LVM provides compatibility with AIX and is similar in functionality to the Linux LVM by also using volume groups and logical volumes.

OS/2 LVM

The OS/2 plug-in provides compatibility with volumes created under OS/2. Unlike the Linux and AIX LVMs, the OS/2 LVM is based on linear linking of disk partitions, as well as bad-block relocation. The OS/2 LVM does not allow for modifications.

Third nested term

The Multi-Disk (MD) plug-in for RAID provides RAID levels linear, 0, 1, 4, and 5 in software. MD is one plug-in that will appear as four region managers you will be able to choose from.

EVMS Features

The next layer is EVMS features. This layer is where new EVMS-native functionality is implemented. EVMS features can be built on any object in the system, including disks, segments, regions, or other feature objects. EVMS features all share a common type of metadata, which makes discovery of feature objects much more efficient, and recovery of broken features objects much more reliable. There are three features currently available in EVMS: Drive Linking, Bad Block Relocation, and Snapshotting.

Nested variable list B.2

First nested term

The Drive Linking feature allows any number of objects to be linearly concatenated together into a single object. A Drive Linked volume can be expanded by adding another storage object to the end or shrunk by removing the last object.

Second nested term

The Bad Block Relocation (BBR) feature monitors its I/O path and detects write failures (which can be caused by a damaged disk). In the event of such a failure, the data from that request is stored in a new location. BBR keeps track of this remapping. Additional I/Os to that location are redirected to the new location.

Third nested term

The Snapshotting feature provides a mechanism for creating a "frozen" copy of a volume at a single instant in time, without having to take that volume off-line. This is useful for performing backups on a live system. Snapshots work

with any volume (EVMS or compatibility), and can use any other available object as a backing store. After a snapshot is created and made into an EVMS volume, writes to the "original" volume cause the original contents of that location to be copied to the snapshot's storage object. Reads to the snapshot volume look like they come from the original at the time the snapshot was created.

Non-nested term.

File System Interface Modules (FSIMs) are the one layer of EVMS that exists only in user space. These plug-ins are used to provide coordination with the filesystems during certain volume management operations. For instance, when expanding or shrinking a volume, the filesystem must also be expanded or shrunk to the appropriate size. Ordering in this example is also important; a filesystem cannot be expanded before the volume, and a volume cannot be shrunk before the filesystem. The FSIMs allow EVMS to ensure this coordination and ordering.

FSIMs also perform filesystem operations from one of the EVMS user interfaces. For instance, a user can make new filesystems and check existing filesystems by interacting with the FSIM.